



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE GRADUAÇÃO
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIAS DA COMPUTAÇÃO

EMILIO CARLOS LOPES DE LEMOS

ESTUDO DE CASO DE IMPLEMENTAÇÃO DE CÁLCULO DE FOLHA DE
PAGAMENTO EM SISTEMA LEGADO USANDO SQL

MOSSORÓ

2017

EMILIO CARLOS LOPES DE LEMOS

ESTUDO DE CASO DE IMPLEMENTAÇÃO DE CÁLCULO DE FOLHA DE
PAGAMENTO EM SISTEMA LEGADO USANDO SQL

Monografia apresentada a Universidade
Federal Rural do Semi-Árido como requisito
para obtenção do título de Bacharel em
Ciências da Computação.

Orientadora: Angélica Félix de Castro, Prof^a.
Dr^a.

MOSSORÓ

2017

© Todos os direitos estão reservados a Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tomar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

L555e LEMOS, EMILIO CARLOS LOPES DE .
ESTUDO DE CASO DE IMPLEMENTAÇÃO DE CÁLCULO DE
FOLHA DE PAGAMENTO EM SISTEMA LEGADO USANDO SQL /
EMILIO CARLOS LOPES DE LEMOS. - 2017.
62 f. : il.

Orientadora: Angélica Félix de Castro.
Monografia (graduação) - Universidade Federal
Rural do Semi-árido, Curso de Ciência da
Computação, 2017.

1. Banco de Dados. 2. Linguagem SQL. 3.
Desempenho. 4. Otimização. I. Castro, Angélica
Félix de , orient. II. Título.

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

EMILIO CARLOS LOPES DE LEMOS

ESTUDO DE CASO DE IMPLEMENTAÇÃO DE CÁLCULO DE FOLHA DE
PAGAMENTO EM SISTEMA LEGADO USANDO SQL

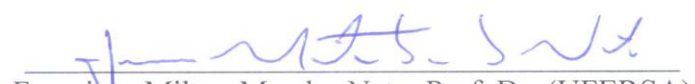
Monografia apresentada a Universidade
Federal Rural do Semi-Árido como requisito
para obtenção do título de Bacharel em
Ciências da Computação.

Defendida em: 17 / 05 / 2017.

BANCA EXAMINADORA


Angélica Félix de Castro, Profª. Drª. (UFERSA)
Presidente


Danniell Cavalcante Lopes, Prof. Dr. (UFERSA)
Membro Examinador


Francisco Milton Mendes Neto, Prof. Dr. (UFERSA)
Membro Examinador

AGRADECIMENTOS

Agradeço a todos aqueles, com muito carinho e apoio, não mediram esforços para que eu chegasse até esta etapa de minha vida.

À professora Angélica Félix, pela paciência na orientação e incentivo que tornaram possível a conclusão desta monografia.

Aos professores da banca examinadora, Danniel Lopes e Milton Mendes, pela oportunidade de fornecer ainda mais conhecimento para a minha formação e por terem aceitado avaliar este trabalho.

A todos os professores do curso, que além de grandes amigos, foram tão importantes na minha vida acadêmica e no desenvolvimento desta monografia.

Aos amigos e colegas, pelo incentivo e pelo apoio constantes.

RESUMO

O desenvolvimento de aplicações empresariais exige soluções que atendam aos requisitos de prazo, desempenho, economia, robustez, escalabilidade e sem o comprometimento da qualidade. Entretanto, atender a todos esses requisitos, simultaneamente, tem sido um grande desafio para as equipes de desenvolvimento. O desempenho da aplicação merece uma atenção especial, pois os usuários estão cada vez mais exigindo que as aplicações operem bem e da forma mais rápida possível. Com base nessa abordagem, este trabalho apresenta o processo de criação de uma nova metodologia para otimizar o cálculo da folha de pagamento de um sistema legado, utilizando primordialmente a linguagem SQL para adquirir o melhor desempenho do servidor de Banco de Dados.

Palavras-chave: Banco de Dados, Linguagem SQL, Desempenho, Otimização.

ABSTRACT

Enterprise application development requires solutions that attend the requirements of term, performance, economy, robustness, scalability and without compromising quality. However, attending all these requirements, simultaneously, has been a major challenge for development teams. Application performance deserves special attention, because the users are increasingly demanding that applications has a good operation, as quickly as possible. Based on this approach, this work presents the process of creating a new methodology to optimize the payroll of a legacy system, using primarily the SQL language to acquire the best performance of the database server.

Keywords: Database, SQL Language, Performance, Optimization.

LISTA DE FIGURAS

Figura 1	– Diagrama de Casos de Uso.....	34
Figura 2	– Diagrama de Entidade e Relacionamento – Módulo I.....	36
Figura 3	– Diagrama de Entidade e Relacionamento – Módulo II.....	37
Figura 4	– Diagrama de Entidade e Relacionamento – Módulo III.....	38
Figura 5	– Diagrama de Entidade e Relacionamento – Módulo IV.....	38
Figura 6	– Diagrama de Entidade e Relacionamento – Módulo V.....	39
Figura 7	– Tela de Listagem e Tela de Cadastro de Empresa.....	41
Figura 8	– Tela de Listagem e Tela de Cadastro de Funcionários.....	42
Figura 9	– Tela de Listagem e Tela de Cadastro de Cargos.....	43
Figura 10	– Tela de Listagem e Tela de Cadastro de Níveis.....	44
Figura 11	– Tela de Listagem e Tela de Cadastro dos valores da Tabela de INSS.....	45
Figura 12	– Tela de Listagem e Tela de Cadastro dos valores da Tabela de IRPF.....	46
Figura 13	– Tela de Listagem e de Cadastro dos valores da Tabela de Salário-Família....	47
Figura 14	– Método Iterativo.....	48
Figura 16	– Diagrama de Fluxo de Dados para o cálculo do Salário Base.....	49
Figura 17	– Comandos SQL para calcular o salário base.....	50
Figura 18	– Diagrama de Fluxo de Dados para o cálculo do desconto do INSS.....	51
Figura 19	– Comandos SQL para calcular o desconto de INSS.....	52
Figura 20	– Diagrama de Fluxo de Dados para o cálculo do desconto de IRPF.....	52
Figura 21	– Comandos SQL para calcular o desconto de IRPF.....	54
Figura 22	– Diagrama de Fluxo de Dados para o cálculo do Salário-Família.....	55
Figura 23	– Comandos SQL para calcular o Salário-Família.....	56

LISTA DE GRÁFICOS

Gráfico 1	– Tempo Médio da folha de pagamento no ambiente local.....	58
Gráfico 2	– Tempo Médio da folha de pagamento no ambiente remoto.....	59

LISTA DE TABELAS

Tabela 1	– Exemplo de uma tabela (relação) funcionário.....	19
Tabela 2	– Tabela do Salário-família 2017.....	30
Tabela 3	– Tabela de INSS 2017.....	30
Tabela 4	– Tabela do IRPF (em vigor desde abril de 2015).....	31

LISTA DE QUADROS

Quadro 1	– Comando SQL para criação de um Banco de Dados.....	20
Quadro 2	– Comando SQL para criação de uma tabela.....	21
Quadro 3	– Comandos de alteração e exclusão de tabela.....	22
Quadro 4	– Sintaxe básica do comando <i>SELECT</i>	23
Quadro 5	– Sintaxe Básica do comando <i>INSERT</i>	23
Quadro 6	– Sintaxe Básica do comando <i>UPDATE</i>	24
Quadro 7	– Sintaxe Básica do comando <i>DELETE</i>	24
Quadro 8	– Exemplos de criação de tabelas temporárias.....	25
Quadro 9	– Fórmula para o cálculo do IRPF.....	32

LISTA DE ABREVIATURAS E SIGLAS

Banco de Dados (BD)

Classificação Brasileira de Ocupação (CBO)

Computer-Aided Software Engineering (CASE)

Data Definition Language (DDL)

Data Manipulation Language (DML)

Diagrama de Entidade e Relacionamento (DER)

Imposto de Renda de Pessoa Física (IRPF)

Instituto Nacional do Seguro Social (INSS)

Linguagem de Modelagem Unificada (UML)

Ministério do Trabalho e Emprego (MTE)

Modelo Entidade-Relacionamento (MER)

Receita Federal do Brasil (RFB)

Sistema de Gerência de Banco de Dados (SGBD)

Structured Query Language (SQL)

SUMÁRIO

1. INTRODUÇÃO	14
1.1 Justificativa	15
1.2. Objetivos.....	15
1.2.1 Objetivo geral	15
1.2.2 Objetivos específicos	15
1.3 Metodologia.....	15
1.4 Estrutura do trabalho	16
2. REFERENCIAL TEÓRICO	17
2.1 Banco de Dados.....	17
2.2 Bancos de Dados Relacionais.....	18
2.3 Linguagem SQL.....	19
2.3.1 Comandos de Definição.....	20
2.3.2 Comandos de Manipulação.....	22
2.4 MySQL 4.1	25
2.4.1. Tabelas temporárias.....	25
2.5 Refatoração e Otimização de comandos SQL	26
2.6 Folha de Pagamento	28
2.6.1 Proventos.....	29
2.6.2 Descontos.....	30
3. ANÁLISES DE PROJETO E SISTEMA	33
3.1 Projeto e Modelagem do Sistema	33
3.2 SisFolha.....	40
3.3 Comparativo das metodologias de cálculo da folha.....	47
3.3.1 Método Iterativo.....	48
3.3.2 Método com Linguagem SQL.....	48
4. COMPARAÇÕES.....	57
4.1 Ambiente de Teste	57
4.1.1 Configurações das Máquinas.....	57
4.2 Resultados.....	57
5. CONCLUSÕES.....	60
REFERÊNCIAS BIBLIOGRÁFICAS	61

1. INTRODUÇÃO

O desenvolvimento de aplicações empresariais exige soluções que atendam aos requisitos de prazo, desempenho, economia, robustez, escalabilidade e sem o comprometimento da qualidade. Entretanto, atender a todos esses requisitos, simultaneamente, tem sido um grande desafio para as equipes de desenvolvimento, principalmente, quando elas têm uma pequena quantidade de integrantes. Geralmente, para atender às demandas dos clientes dentro dos prazos contratados, os desenvolvedores utilizam linguagens de programação orientadas a objetos e ferramentas de aumento de produtividade, como ambientes de programação gráficos do tipo “arrastar e soltar”, além de *frameworks* que abstraem o acesso aos dados.

Dentre os requisitos já citados, o desempenho da aplicação merece uma atenção especial, pois em muitos casos o usuário não estará disposto a esperar vários minutos para visualizar um relatório importante, por exemplo. Uma situação como esta pode ser fator decisivo para a compra ou rejeição do *software*, especialmente, quando se trata de manipulação de informações estratégicas armazenadas em bancos de dados.

Para melhorar o tempo de resposta de uma aplicação, as empresas adquirem computadores mais velozes, mudam o banco de dados, e refatoram seus sistemas, proporcionando melhoria na performance da aplicação.

É muito comum que a maioria dos esforços estejam concentrados longe de onde residem os dados, isto é, as regras de negócios são definidas em *frameworks*, que estarão executando em algum lugar, seja na estação do usuário ou num servidor de aplicação intermediário que realizará consultas à bases de dados remota. Embora os *frameworks* proporcionem um aumento de produtividade e permitam uma maior portabilidade e independência da base de dados, nem sempre eles podem garantir um bom desempenho da aplicação.

Diante de uma situação como o cálculo de uma folha de pagamento, é possível transferir a maioria dos cálculos para o próprio servidor de banco de dados, caso este possua facilidades como *stored procedures* e *triggers*. Entretanto, caso o banco utilizado não ofereça tais funcionalidades, a lógica ficará mantida dentro da aplicação, que realizará diversas consultas aos dados para obter os resultados esperados, através do uso bem planejado de instruções do tipo *Structured Query Language* (SQL).

1.1 Justificativa

A escolha do tema se deu pelo interesse em demonstrar a resolução de problemas envolvendo bancos de dados usando a linguagem SQL, sem a necessidade de implementação de *stored procedures*.

1.2. Objetivos

Neste tópico, serão apresentados os objetivos gerais e específicos a serem realizados nesta pesquisa.

1.2.1 Objetivo geral

O objetivo deste trabalho é desenvolver as principais rotinas de cálculo de uma folha de pagamento em um sistema legado usando apenas instruções SQL, mantendo o tempo do processamento aceitável dentro da faixa de alguns segundos, observando os resultados para quantidades diferentes de funcionários.

1.2.2 Objetivos específicos

Os objetivos específicos deste trabalho são:

- Apresentar a modelagem do projeto de *software* e as principais funcionalidades do sistema de folha de pagamento utilizado como base para esta pesquisa;
- Demonstrar o fluxo básico das principais rotinas de cálculo de uma folha de pagamento, tais como: cálculo de salário base, imposto de renda, previdência social e salário família;
- Demonstrar as operações ditas anteriormente usando apenas instruções SQL;

1.3 Metodologia

A proposta deste trabalho consiste em desenvolver um sistema legado de folha de pagamento que irá realizar os cálculos relativos à remuneração, proventos e descontos dos funcionários de uma determinada empresa, utilizando a linguagem SQL. A partir disso, realizar um estudo de caso, traçando um comparativo de desempenho para diferentes quantidades de funcionários.

1.4 Estrutura do trabalho

No segundo capítulo deste trabalho, será apresentado o Referencial Teórico, em que são explanadas abordagens importantes sobre Bancos de Dados, características de Bancos de Dados Relacionais e Linguagem SQL, Refatoração e Otimização de comandos SQL, além de alguns aspectos sobre Folha de Pagamento.

No terceiro capítulo, explana-se, primeiramente, as principais funcionalidades do sistema. Logo depois, explica-se o desenvolvimento do cálculo da Folha de Pagamento usando a linguagem SQL.

No quarto capítulo, realiza-se uma demonstração dos resultados de cálculo de folha de pagamento realizados para diferentes quantidades de funcionários. Por fim, no quinto capítulo, são apresentadas as conclusões adquiridas nesta pesquisa.

2. REFERENCIAL TEÓRICO

Neste tópico, serão analisados os conceitos sobre Bancos de Dados, Bancos de Dados Relacionais, uma abordagem sobre a Linguagem SQL, também serão abordados os conceitos de Refatoração e Otimização de comandos SQL, e ainda serão explanadas algumas definições a respeito de Folha de Pagamento.

2.1 Banco de Dados

Existem inúmeras definições sobre Banco de Dados (BD). De acordo com Date (2004), um BD é uma coleção de elementos persistentes, usada pelos sistemas de aplicação de uma determinada organização comercial, científica, técnica ou autônoma.

Já para Machado (2008), um Banco de Dados possui as seguintes propriedades:

- É uma coleção lógica coerente de dados com um significado específico; desde modo, uma disposição desordenada dos dados não pode ser referenciada como Banco de Dados;
- Ele é projetado, construído e preenchido com valores de dados para um propósito pertinente; um Banco de Dados possui um conjunto predefinido de usuários e de aplicações;
- Ele apresenta algum aspecto do mundo real, o qual é chamado de minimundo; qualquer alteração efetuada no minimundo é automaticamente refletida no Banco de Dados.

Por outra perspectiva, Heuser (2009) afirma que o conjunto de arquivos integrados que atendem a um conjunto de sistemas recebe o nome de Banco de Dados. Para o autor, o compartilhamento de dados tem reflexos na estrutura do *software*: a estrutura interna dos arquivos passa a ser mais complexa, pois estes devem ser construídos de forma a atender às necessidades dos diferentes sistemas. Para contornar este problema, usa-se um Sistema de Gerência de Banco de Dados (SGBD), um *software* que incorpora as funções de definição, recuperação e alteração de dados em um Banco de Dados.

Antes de criar um BD, é necessário, primeiramente, elaborar seu projeto (LINS, 2016). O objetivo principal da modelagem de dados é transmitir e apresentar uma representação única, não redundante e também resumida, dos dados de uma aplicação. Em projetos conceituais de aplicações de Bancos de Dados o modelo Entidade-Relacionamento é o mais utilizado para representação e entendimento dos dados que compõem a essência de um sistema (MACHADO, 2008).

Existem alguns paradigmas para a criação de Bancos de Dados. Dentre esses paradigmas, conforme já foi dito anteriormente, o mais utilizado é o Modelo Entidade-Relacionamento. Este será o modelo adotado para esta pesquisa. No próximo tópico, é apresentada uma abordagem sobre este modelo.

2.2 Banco de Dados Relacionais

Os conceitos do Modelo Entidade-Relacionamento (MER), segundo Machado (2008), estão destinados, prioritariamente, ao projeto de Banco de Dados, entretanto eles podem ser utilizados para o entendimento de um determinado negócio, por exemplo, na modelagem de processos de negócios, bem como auxiliar o desenvolvimento de estruturas de dados que possam ser implementadas fora de um ambiente de armazenamento de dados, juntamente com alguma linguagem de programação.

Informalmente, Date (2004), afirma que o MER é constantemente descrito como tendo os três aspectos a seguir:

- Aspecto estrutural: os dados de um Banco de Dados são percebidos pelo usuário como tabelas;
- Aspecto de integridade: essas tabelas satisfazem a certas restrições de integridade;
- Aspecto manipulador: os operadores disponíveis para que o usuário possa manipular essas tabelas são de restrição, junção e projeção.

Formalmente falando, Date (2004) constata que o MER consiste nos cinco componentes a seguir:

- Uma coleção ilimitada de tipos escalares;
- Um gerador de tipo de relação e uma interpretação pretendida para esses dois tipos de relações;
- Recursos para definição de *RelVars* (variável de relação) desses tipos de relações;
- Um operador de atribuição relacional para concessão de valores de relações a essas *RelVars*;
- Uma coleção ilimitada de operadores relacionais genéricos para derivar valores de relações a partir de outras relações.

Na terminologia formal do MER, de acordo com Elmasri e Navathe (2011), na tabela, uma linha é chamada de tupla, um cabeçalho da coluna é chamado de atributo, e a

tabela estabelece uma relação. O tipo de dado que descreve os valores de cada coluna é representado por um domínio de valores possíveis.

Para exemplificar uma tabela de Banco de Dados, a Tabela 1 mostra um esquema representando um cadastro de funcionários, no qual são definidos os seguintes atributos: matrícula, nome, salário, telefone.

Tabela 1 - Exemplo de uma tabela (relação) funcionário.

Matrícula	Nome	Salário	Telefone
000001	Ângelo Cláudio	1000,00	(84) 8888-8888
000002	Francisco José	1000,00	(84) 8877-8877
000003	João Carlos	2000,00	(84) 8866-8866
000004	Larissa Silva	2000,00	(84) 8855-8855

Fonte: Autoria Própria.

Conforme mostra a primeira linha da tabela, pode-se concluir que existe um funcionário de nome Ângelo Cláudio, de matrícula 000001, com salário de R\$ 1000,00 e telefone (84) 8888-8888. Então, percebe-se que cada linha de uma tabela representa uma entidade, um assunto que é descrito pelos valores de cada uma das colunas.

De maneira mais direta, o MER baseia-se na percepção de um universo constituído por um grupo de objetos chamados entidades e por relacionamentos entre eles. Esse modelo foi desenvolvido para facilitar o projeto de Banco de Dados permitindo a especificação de um esquema de empreendimento. Tal esquema representa a estrutura lógica global do Banco de Dados (KORTH e SILBERSCHATZ, 2006).

No tópico a seguir, será realizada uma análise sobre a linguagem SQL que mostra os principais comandos utilizados para a implementação desta pesquisa.

2.3 Linguagem SQL

A linguagem SQL foi desenvolvida originalmente pela *IBM Research* no início da década de 1970. Sua sigla tem como significado: *Structured Query Language* – Linguagem Estruturada de Pesquisa. Devido ao sucesso sobre como a linguagem SQL consulta e manipula os dados dentro de um ambiente de Banco de Dados, a sua utilização cresceu rapidamente. Assim, essa linguagem se tornou um padrão de fato no mundo dos Bancos de Dados Relacionais (MACHADO, 2008).

De acordo com Date (2004), a SQL possui um componente de linguagem de definição de dados (*Data Definition Language* - DDL) e um componente de linguagem de manipulação de dados (*Data Manipulation Language* - DML).

A DDL permite ao usuário a definição da estrutura e organização dos dados armazenados, e as relações que existem entre eles. Já DML, permite ao usuário ou a um programa de aplicação a inclusão, remoção, seleção ou atualização de dados previamente armazenados no banco (MACHADO, 2008).

Algumas vantagens da linguagem SQL citadas por Machado (2004) são:

- Independência de fabricante;
- Portabilidade entre computadores;
- Redução dos custos com treinamento;
- Inglês estruturado de alto nível;
- Consulta Interativa;
- Múltiplas visões dos dados;
- Definição dinâmica dos dados.

No próximo tópico, têm-se alguns dos principais comandos utilizados pela linguagem SQL, tanto de definição quanto de manipulação de dados.

2.3.1 Comandos de Definição

Em um Banco de Dados é possível realizar operações para a criação e distribuição de tabelas de uma determinada aplicação, dependendo do ambiente de SGBD utilizado (MACHADO, 2008). O Quadro 1 exibe, como exemplo, o comando SQL para a criação de um banco intitulado “*folha*”.

Quadro 1 – Comando SQL para criação de um Banco de Dados.

Definição:

```
CREATE DATABASE nome_do_banco;
```

Exemplo:

```
CREATE DATABASE folha;
```

Fonte: Adaptado de Machado (2008).

O comando *CREATE DATABASE* é responsável pela criação de uma base de dados. Uma vez que o banco foi criado, pode-se dar início a criação das tabelas. O Quadro 2 mostra, como exemplo, o comando para geração de uma tabela.

Quadro 2 – Comando SQL para criação de uma tabela.

Definição:

CREATE TABLE nome_da_tabela (definição das colunas);

Exemplo 1:

CREATE TABLE funcionarios (codigo int, nome varchar(50), telefone varchar(14));

Exemplo 2:

CREATE TEMPORARY TABLE pessoas (codigo int, nome varchar(50));

Fonte: Adaptado de Machado (2008).

O comando *CREATE TABLE* é responsável pela criação de uma tabela. A definição das colunas (campos) da tabela ocorre durante a sua criação. O exemplo 1 do Quadro 2 mostra a criação da tabela intitulada “*funcionarios*”, que possui a definição dos campos “*codigo*”, “*nome*” e “*telefone*”. O primeiro campo é do tipo inteiro (*int*), o segundo é do tipo caractere (*varchar*) com tamanho máximo de 50 caracteres, e o último, também do tipo caractere, com tamanho máximo de 14 caracteres.

O exemplo 2 do Quadro 2 mostra a criação da tabela temporária intitulada “*pessoas*”. De acordo com Ferrari (2017), uma tabela temporária possui “vida útil” de uma sessão ou transação. Ela está vazia quando a sessão ou transação começa e descarta os dados. Uma tabela temporária é associada à transação. Isto significa que ao término da transação os dados da tabela são perdidos, porém sua descrição permanece gravada no banco de dados mesmo após a mudança de sessão.

É possível realizar alguma alteração (*ALTER TABLE*) em uma determinada coluna na tabela já criada. Também é admissível realizar a exclusão (*DROP TABLE*) de tabelas. Essas operações também são comandos de definição e são exemplificadas no Quadro 3, logo a seguir.

Quadro 3 – Comandos de alteração e exclusão de tabela.

Definição do comando de alteração para adicionar uma nova coluna na tabela:

ALTER TABLE nome_da_tabela ADD nome_do_campo TIPO;

Exemplo:

ALTER TABLE funcionarios ADD cpf varchar(12);

Definição do comando de alteração para excluir uma coluna na tabela:

ALTER TABLE nome_da_tabela DROP COLUMN nome_do_campo;

Exemplo:

ALTER TABLE funcionarios DROP COLUMN cpf;

Definição do comando de exclusão:

DROP TABLE nome_da_tabela;

Exemplo:

DROP TABLE funcionários;

Fonte: Adaptado de Machado (2008).

Esses foram os principais comandos de definição usados nesta pesquisa. No próximo tópico, serão apresentados os comandos de manipulação de dados, explanando suas definições e mostrando alguns exemplos básicos.

2.3.2 Comandos de Manipulação

Os comandos de manipulação de dados permitem que o usuário possa realizar os comandos de inclusão, remoção, seleção ou atualização de dados previamente armazenados na base. Uma das operações mais comuns é examinar as informações armazenadas. Estas operações são realizadas pelo comando *SELECT* (MACHADO, 2008).

Segundo Machado (2008), o *SELECT* tem palavras-chave em um comando básico:

- *SELECT*: especifica as colunas da tabela que queremos selecionar;
- *FROM*: especifica as tabelas;
- *WHERE*: especifica a condição de seleção das linhas.

O Quadro 4 mostra a definição e exemplifica o comando de seleção a partir de sua sintaxe básica.

Quadro 4 – Sintaxe básica do comando *SELECT*.

Definição:

SELECT <nome(s) da(s) coluna(s)> FROM <tabelas>;

Exemplo 1:

SELECT nome FROM funcionarios;

Exemplo 2:

SELECT nome FROM funcionarios WHERE telefone = 88888888;

Fonte: Adaptado de Machado (2008).

A execução do comando do exemplo 1 exibirá o nome de todos os funcionários inseridos na tabela requerida, uma vez que não apresenta nenhuma condição para a projeção dos dados. Já o exemplo 2, mostrará os funcionários que tem como número de telefone o conjunto de caracteres “88888888”.

Os próximos comandos a serem explanados são a respeito das operações de inserção, atualização e deleção de registro em uma determinada tabela do BD. Para inserir registros em uma tabela utiliza-se o comando *INSERT*. A sintaxe básica e o exemplo desse comando são exibidos no quadro 5, a seguir.

Quadro 5 – Sintaxe Básica do comando *INSERT*.

Definição:

INSERT INTO nome_da_tabela (nomes_das_colunas) VALUES (valores);

Exemplo 1:

*INSERT INTO funcionarios (codigo, nome, telefone)
VALUES (“000001”, “Maria”, “88888888”);*

Exemplo 2:

*INSERT INTO funcionarios (codigo, nome, telefone)
SELECT codigo, nome, telefone FROM pessoas WHERE codigo = “000010”;*

Fonte: Adaptado de Machado (2008).

A cláusula *VALUE* especifica os dados a serem inseridos na tabela. Ela é uma palavra requerida para introduzir no comando uma lista de valores para cada coluna (MACHADO,

2008). A execução do comando do exemplo 1 terá como resultado a inserção do registro na tabela “*funcionarios*”, que terá o código igual a “0000001”, o nome “Maria” e o telefone “88888888”. Já no exemplo 2, a inserção ocorre juntamente com o comando *SELECT*. Assim, o resultado da busca na tabela “*peessoas*” é inserido na tabela “*funcionarios*”.

A atualização de dados em linhas existentes na tabela permite que seja especificada uma determinada coluna e altere-se seu valor (MACHADO, 2008). Para alterar registros de uma tabela utiliza-se o comando *UPDATE*. A sintaxe básica e o exemplo dessa operação são exibidos no Quadro 6, a seguir.

Quadro 6 – Sintaxe Básica do comando *UPDATE*.

Definição:

UPDATE nome_da_tabela SET nome_da(s)_coluna(s) = valor

WHERE condição;

Exemplo:

UPDATE funcionario SET telefone = “99999999” WHERE código = “000001”;

Fonte: Adaptado de Machado (2008).

A execução do comando do exemplo acima atualiza o telefone do funcionário de código “000001” para “99999999”.

Para apagar registros de uma tabelas utiliza-se o comando *DELETE*. A sintaxe básica e o exemplo dessa operação são apresentados no Quadro 7, logo abaixo.

Quadro 7 – Sintaxe Básica do comando *DELETE*.

Definição:

DELETE FROM nome_da_tabela WHERE condição;

Exemplo:

DELETE FROM funcionarios WHERE codigo = “000002”;

Fonte: Adaptado de Machado (2008).

O comando do exemplo acima exclui da tabela “*funcionarios*” o registro que tem o código “000002”.

Existem vários outros comandos de definição em SQL que pode ser utilizado pelo desenvolvedor, conforme a sua necessidade. Para isso, é interessante a realização de um estudo mais aprofundado nesta vasta linguagem, a fim de saber como usá-la adequadamente.

2.4 MySQL 4.1

O MySQL 4.1 é o Banco de Dados adotado pelo cliente da aplicação desta pesquisa. Ele é um servidor de bancos de dados SQL rápido e robusto, multitarefa e multiusuário, podendo ser usado em sistemas de produção com alta carga e missão crítica, e também pode ser embutido em programa de uso em massa (MYSQL, 2017).

Apesar de apresentar as vantagens ditas acima, esta versão não permite criar *Stored Procedures*, que poderiam ser utilizadas para realizar os cálculos da folha de pagamento. No entanto, este banco permite criar tabelas temporárias, que podem ser uma alternativa para realizar o cálculo da folha.

2.4.1 Tabelas Temporárias

São aquelas criadas internamente pelo *MySQL*, com espaço em memória, para resolver uma consulta que conte com as seguintes cláusulas: *UNION*, *ORDER BY*, *GROUP BY*, *DISTINCT*. Tabelas temporárias existem somente dentro da sessão atual e são automaticamente destruídas no final da sessão. Elas são criadas na memória do cliente e não são compartilhadas como as demais tabelas, isto é, essas tabelas só são visíveis dentro da sessão e da estação na qual foram criadas. Sendo assim, duas sessões diferentes podem criar as mesmas tabelas temporárias sem que haja qualquer conflito entre elas.

Tabelas temporárias podem ser criadas da maneira tradicional com todos os campos sendo definidos, ou dinamicamente, por meio do comando *SELECT*. A mesma tabela temporária não pode ser selecionada mais de uma vez dentro de um comando *SELECT*. O quadro abaixo mostra exemplos para as situações ditas anteriormente.

Quadro 8 – Exemplos de criação de tabelas temporárias.

```
CREATE TEMPORARY TABLE tmp_funcionario
SELECT a.*, b.vl_cargo from funcionario a
LEFT JOIN cargo b on b.cd_cargo=a.cd_cargo;

SELECT * from tmp_funcionario a where cd_cargo = 01
UNION ALL
SELECT * from tmp_funcionario b where cd_cargo = 02;
```

Fonte: Autoria própria.

As tabelas temporárias são usadas nesta pesquisa como uma espécie de rascunho para o cálculo da folha, acumulando os resultados em cada passo do cálculo.

2.5 Refatoração e Otimização de comandos SQL

A refatoração consiste em uma técnica de programação que trata da recodificação de um projeto de *software* de forma disciplinada. A ideia básica é realizar pequenas mudanças no código fonte a fim de melhorar o projeto, facilitar o entendimento e manutenção. Refatoração possibilita a evolução gradativa do código ao longo do tempo, adotando uma abordagem iterativa e incremental para a programação (DIAS NETO, 2017).

Além da refatoração, outra técnica que contribui positivamente para o desempenho de um determinado sistema, é a otimização de consultas SQL. De acordo com Santos (2017), fazer o aprimoramento de uma consulta é mais uma arte do que uma ciência. Uma mesma pesquisa pode ter um comportamento totalmente diferente mediante ao montante de registros existente, às chaves presentes, aos recursos do sistema, às estatísticas de dados, aos índices e outros fatores.

Para realizar o aprimoramento de consultas SQL é importante considerar cada uma caso a caso e tentar uma variedade de técnicas de otimização.

Alguns cuidados precisam ser tomados ao escrever uma busca SQL, dentre eles, segundo Santos (2017), tem-se:

1. Favorecimento à lógica de conjuntos ao invés da lógica de procedimentos
 - O fator mais importante a ser considerado quando estiver otimizando *queries* é saber como funciona a manipulação da lógica de conjuntos de registros;
 - Cursores e outras construções procedimentais geralmente limitam a habilidade do otimizador de consultas para gerar planos de consultas flexíveis.
2. Teste de variações de consultas objetivando a performance
 - Frequentemente o otimizador pode produzir planos totalmente diferentes para *queries* logicamente equivalentes;
 - Testar técnicas diferentes, como *joins* ou *subqueries*, para descobrir qual delas é mais eficiente.
3. Evitar *QUERY HINTS*

- *Query hints* diz ao otimizador como se comportar, portanto, sobrescreve a habilidade do otimizador de fazer o seu trabalho adequadamente. Eliminando as escolhas do otimizador, ocorrerá uma limitação a um plano que é menos eficaz que o ideal;
- Essa técnica deve ser utilizada somente quando tem-se absoluta certeza de que o otimizador está incorreto na sua escolha perante a lógica do seu negócio.

4. Utilização de subconsultas correlatas

- O otimizador está apto a trabalhar para integrar *subqueries* no fluxo da *query* principal em uma variedade de caminhos escolhidos por ele;
- As subconsultas podem ser úteis em diversas situações para o ganho de performance, por exemplo, ao realizar um *JOIN* para uma tabela apenas para verificar a existência de linhas correlatas;
- Para uma melhor performance, deve-se trocar estes tipos de *JOINS* por queries correlatas que fazem uso do operador *EXISTS*.

5. Evitar funções definidas pelo usuário na cláusula *WHERE*

- Funções definidas pelo usuário que retornam apenas um valor, diferentemente das *subqueries* escalares, não são otimizadas no plano principal da *query*;
- Usar função escalar na lista do *SELECT* é muito menos problemático porque as linhas já foram filtradas na cláusula *WHERE*.

6. Utilização funções tabulares como tabelas derivadas

- São funções que retornam o resultado em formato de tabela;
- Ao contrário do item cinco, funções tabulares são geralmente úteis, no ponto de vista de performance, quando são utilizadas como tabelas derivadas;
- O otimizador avalia uma tabela derivada apenas uma vez por consulta. Se uma função definida pelo usuário em uma tabela derivada tiver uma lógica que atenda outras consultas, pode-se, então, encapsular e reutilizá-la para outras *queries*.

7. Evitar colunas *GROUP BY* desnecessárias

- Quanto mais colunas na lista da cláusula *GROUP BY* for adicionada, mais o processo de agrupar linhas torna-se dispendioso;
- Se a consulta possui poucas colunas de agregação e muitas colunas agrupadas não agregadas, deve-se pensar em reconstruí-la utilizando *subqueries* escalares correlatas. Isso resultará em menos trabalho para agrupar na consulta.

8. Uso de expressões *CASE*

- A expressão *CASE* é uma das mais poderosas ferramentas de lógica disponível para os programadores SQL. Utilizando essa expressão pode-se, por exemplo, mudar dinamicamente a saída de uma coluna;
- Isso habilita a consulta a retornar somente os dados que são absolutamente necessários, reduzindo assim operações de entrada e saída e rede, ao invés de gerar um *RESULT SET* muito grande e desnecessário ao cliente.

9. Dividir *JOINS* em tabelas temporárias

- A estratégia principal do otimizador é encontrar planos que satisfaçam consultas utilizando operações simples;
- Embora essa estratégia funcione para a maioria dos casos, ela pode falhar quando a saída de dados for grande, pois muitos *joins* requerem muita operação de entrada e saída;
- Em alguns casos, a melhor opção é reduzir o trabalho gerando uma tabela temporária preenchida com parte do que se deseja da consulta original, unindo colunas que estavam em tabelas distintas, para diminuir a quantidade de *JOINS*. Pode-se então fazer uma junção com tabelas temporárias para produzir um resultado final;
- Essa técnica não é muito favorável em sistemas transacionais pesados porque vai ocorrer uma sobrecarga de criação dessas tabelas temporárias, mas isso pode ser muito útil em situações de tomada de decisão (*Business Intelligence*).

2.6 Folha de Pagamento

As empresas que assumem o risco de atividade econômica urbana ou rural, com fins lucrativos ou não, estão obrigadas a preparar mensalmente a folha de pagamento da

remuneração paga ou creditada a todos os segurados a seu serviço, devendo manter em cada estabelecimento, uma via desta (GONÇALVES, 2007).

Segundo Gonçalves (2007), tal obrigatoriedade está prevista no art. 225 do Dec. 3.048/99, que ainda impõe que este documento deverá discriminar:

- O nome de todos os segurados, indicando cargo, função ou serviço prestado;
- Agrupar os segurados por categoria, assim entendido:
 - Segurado empregado;
 - Trabalhador avulso;
 - Contribuinte individual;
- Destacar o nome dos segurados e gozo de salário-maternidade;
- Destacar as parcelas integrantes e não-integrantes da remuneração;
- Descontos legais;
- Indicar o número de cotas de salário-família atribuídas a cada segurado ou trabalhador avulso.

A Folha de Pagamento é constituída de proventos e descontos. Os principais proventos a serem estudados nesta pesquisa são: salário base e salário-família. Já os descontos estudados serão: previdência social (INSS) e imposto de renda (IRRF).

2.6.1 Proventos

O salário base, também chamado de salário contratual, é a contraprestação devida e paga diretamente pelo empregador a todo empregado. Ele pode ser pago mensal, quinzenal, semanal ou diariamente, por peça ou tarefa; o salário nunca poderá ser inferior ao salário mínimo; é o valor utilizado normalmente como base para os cálculos dos proventos e descontos mensais (OLIVEIRA, 2007).

Já o salário-família, é um valor pago ao empregado e ao trabalhador avulso, de acordo com o número de filhos ou equiparados que possua. Filhos maiores de quatorze anos não têm direito, exceto no caso dos inválidos (para quem não há limite de idade). Para ter direito ao salário-família, o cidadão precisa enquadrar-se no limite máximo de renda estipulado pelo governo federal (PREVIDÊNCIA, 2017).

O valor do salário-família é reajustado periodicamente, por isso, antes de pagá-lo, convém certificar-se do valor. A Tabela 2 mostra os valores referentes ao salário-família.

Tabela 2 – Tabela do Salário-família 2017.

Vigência	Remuneração	Salário-família
A Partir de 01/01/2017	R\$ 859,88	R\$ 44,09
	R\$ 859,89 a R\$ 1.292,43	R\$ 31,07

Fonte: Guia trabalhista (2017).

A tabela acima está em vigor desde janeiro de 2017. Ela informa que os funcionários que têm remuneração de até R\$ 859,88 possuem salário-família de R\$ 44,09. E os funcionários que têm remuneração entre R\$ 859,89 e R\$ 1.292,43 possuem salário-família de R\$ 31,07.

2.6.2 Descontos

O Instituto Nacional do Seguro Social (INSS) é responsável pelo pagamento da aposentadoria e demais benefícios aos trabalhadores brasileiros, com exceção de servidores públicos. Para ter direito ao benefício, o trabalhador deverá pagar uma contribuição mensal durante um determinado período ao INSS, variando de acordo com o tipo de aposentadoria (CAIXA, 2017). Esse valor é descontado do salário base do funcionário todo mês, e é calculado mediante a aplicação da correspondente alíquota, de forma não cumulativa, sobre seu salário de contribuição, de acordo com a tabela estabelecida periodicamente pelo Governo Federal.

A Tabela 3 exibe os limite de valores de salários e as respectivas alíquotas a serem descontadas dos salários dos funcionários.

Tabela 3 – Tabela de INSS 2017.

Tabela para Empregado, Empregado Doméstico e Trabalhador Avulso 2017	
Salário de Contribuição (R\$)	Alíquota
Até R\$ 1.659,38	8%
De R\$ 1.659,39 a R\$ 2.765,66	9%
De R\$ 2.765,67 até R\$ 5.531,31	11%

Fonte: Previdência (2017).

Um exemplo de aplicação da tabela acima pode ser apresentado da seguinte forma: um empregado que tem como salário de contribuição um valor equivalente a R\$ 1200,00, sofrerá um desconto, na folha de pagamento, de R\$ 96,00, ou seja, 8% de R\$ 1200,00. Outro exemplo seria: um empregado que tem como salário de contribuição um valor equivalente a

R\$ 8000.00, sofrerá um desconto, na folha de pagamento, de R\$ 608.44, ou seja, 11% de R\$ 5531.31, que é o limite.

Assim, enquanto vigorar a tabela acima, ninguém sofrerá desconto previdenciário superior a R\$ 608.44, por mais alto que seja seu salário (GONÇALVES, 2007).

Já o desconto de imposto de renda (IRRF) é um dos tributos administrados pela Receita Federal do Brasil (RFB), sendo devido tanto pelas pessoas jurídicas quanto físicas. Os contribuintes do Imposto de Renda Pessoa Física (IRPF) são as pessoas físicas domiciliadas ou residentes no Brasil, titulares de disponibilidade econômica ou jurídica de renda ou proventos de qualquer natureza, inclusive rendimentos e ganhos de capital, sem distinção da nacionalidade, sexo, idade, estado civil ou profissão (PORTAL TRIBUTÁRIO, 2017). O desconto do IRPF é calculado mediante a aplicação das correspondentes alíquotas que constam na tabela também estabelecida periodicamente pelo Governo Federal, sobre a somatória das verbas recebidas pelo empregado, que possuem caráter salarial.

A Tabela 4 mostra a base de cálculo, as respectivas alíquotas a serem descontadas dos salários dos funcionários e as parcelas a deduzir do IRPF.

Tabelas 4 – Tabela do IRPF (em vigor desde abril de 2015).

Base de cálculo (R\$)	Alíquota (%)	Parcela a deduzir do IRPF (R\$)
Até 1.903,98	-	-
De 1.903,99 até 2.826,65	7,5	142,80
De 2.826,66 até 3.751,05	15	354,80
De 3.751,06 até 4.664,68	22,5	636,13
Acima de 4.664,68	27,5	869,36

Fonte: Receita (2017).

Sem ainda considerar as deduções, tem-se como exemplo: um empregado que recebe uma remuneração equivalente a R\$ 2400.00 sofrerá um desconto de R\$ 37.20, ou seja: $(R\$ 2400.00 \times 7.5\%) - R\$ 142.80 = R\$ 37.20$.

Porém, segundo Machado (2008), após apurar a somatória do salário do empregado e antes de ser aplicada a tabela de IR, pode-se efetuar algumas deduções, que são:

- R\$ 189.59 por dependente (valor para 2017);
- O valor já descontado de INSS;
- Pensão alimentícia;
- Contribuição para as entidade de previdência privada.

Agora que tem-se o critério da deduções, pode-se recalcular o último exemplo, para realmente obter o desconto de IR. A fórmula do Quadro 9 torna mais clara a demonstração para o cálculo do IR:

Quadro 9 – Fórmula para o cálculo do IRPF.

$\text{IR} = \{[(\text{salário} - \text{deduções}) \times \text{alíquota}] - \text{parcela a deduzir}\}$
--

Fonte: Machado (2008).

Como demonstração de aplicação da fórmula acima, tem-se o seguinte exemplo:

- Salário: R\$ 2400.00;
- Dependentes: 1 (R\$ 189.59 por dependente);
- Desconto INSS: R\$ 198.00 (9% de R\$ 2400.00);
- Cálculo do IR = $\{[(\text{R\$ } 2400.00 - (189.59 + 216.00)) \times 7.5\%] - 142.80\}$;
- IR = R\$ 6.78.

Assim, pode-se concluir que o valor de imposto de renda a ser descontado do salário do empregado (R\$ 2400.00) é de R\$ 6.78.

No capítulo a seguir, será apresentado o projeto para o desenvolvimento do sistema de folha de pagamento, também serão explanadas as principais funcionalidades e será mostrado um comparativo em relação às metodologias utilizadas para realizar o cálculo da folha de cada funcionário.

3. ANÁLISES DE PROJETO E SISTEMA

As principais análises realizadas nesta pesquisa contemplam a modelagem do sistema, como também suas principais funcionalidades e ainda a metodologias utilizadas para a realização do cálculo da folha para quantidades diferentes de funcionários.

3.1 Projeto e Modelagem do Sistema

Antes de desenvolver o sistema deste trabalho foi necessário projetá-lo, ou seja, modelar todas as suas funcionalidades. Esta modelagem foi feita com base nas práticas oferecidas pela Engenharia de *Software*. As práticas utilizadas foram: Especificação de Requisitos, Diagramas de Casos de Uso e Diagrama de Entidade e Relacionamento. As especificações de cada uma destas práticas são explanadas detalhadamente.

- Especificação de Requisitos

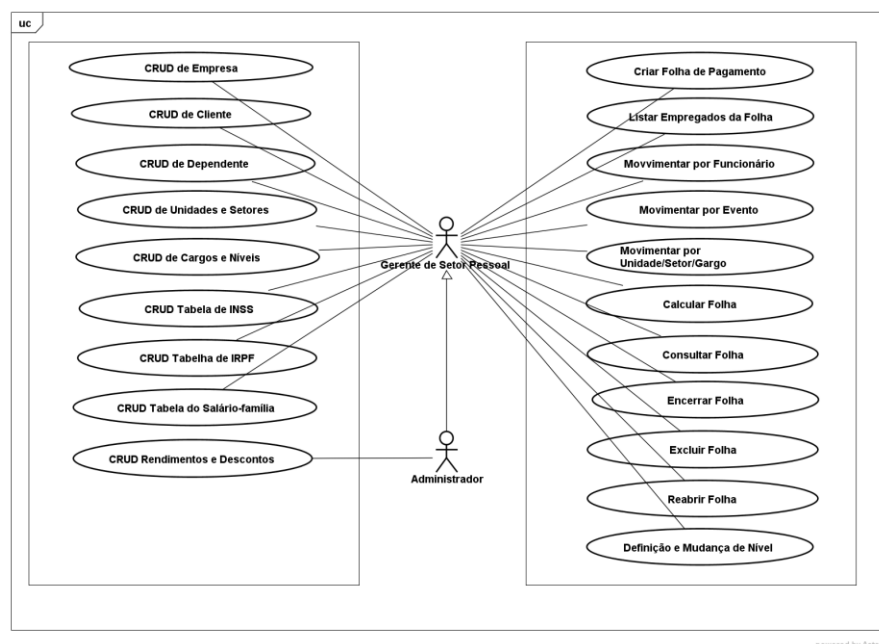
Na fase da especificação de requisitos, o objetivo principal é analisar as necessidades, avaliar a possibilidade de desenvolvimento computacional, propor uma solução e validar a especificação. O *software* desta pesquisa tem por objetivo gerenciar as informações relevantes dos funcionários de uma determinada empresa e realizar o cálculo da folha de pagamento. Este sistema deve ser acessível ao responsável pelo gerenciamento do setor pessoal da empresa em questão. Trata-se de um sistema *desktop*, ou seja, funciona apenas no computador em que ele foi instalado; o Banco de Dados desta aplicação pode ser instalado localmente ou ainda via rede, por meio de outra máquina utilizada como servidor de dados. A seguir, os requisitos do sistema serão listados.

- Requisitos funcionais de inserção, exclusão, atualização, listagem e busca para:
 - Empresa;
 - Funcionário;
 - Dependentes;
 - Unidades e setores;
 - Cargos e níveis;
 - Tabela de INSS;
 - Tabela de IRPF;
 - Tabela do Salário-família;
 - Rendimentos e Descontos.

- O sistema também tem como requisitos:
 - Criar Folha de Pagamento;
 - Listar Empregados da Folha;
 - Realiza Movimentações;
 - Por funcionário;
 - Por evento;
 - Por unidade, setor e cargo;
 - Calcular Folha de Pagamento;
 - Consultar Folha de Pagamento;
 - Encerrar Folha;
 - Excluir Folha;
 - Reabrir Folha;
 - Definição e Mudança de Nível.
- Diagramas de Casos de Uso

Os Casos de Uso descrevem as funcionalidades que serão construídas no sistema proposto e também a sequência de eventos de um ator que usa a aplicação para completar um dos processos. A Figura 1 apresenta o Diagrama de Casos de Uso relacionados aos atores Gerente de Setor Pessoal e Administrador.

Figura 1 – Diagrama de Casos de Uso.



O diagrama acima, modelado a partir da Linguagem de Modelagem Unificada (UML) e a ferramenta CASE *Astah Community*, tem por objetivo descrever uma visão externa do sistema e representar as relações entre os atores Administrador e Gerente de Setor Pessoal, e os casos de uso a eles associados. Os atores são representados pelos bonecos localizados ao centro da figura. Cada Caso de Uso é representado pelas elipses que estão nas laterais da imagem. As linhas que ligam os atores aos Casos de Uso representam a comunicação entre eles, o que significa que cada ator interage com o sistema, trocando informações por meio desses Casos de Uso. A linha com uma seta que liga os dois atores representa um relação de herança, ou seja, o ator Administrador herda todos os Casos de Uso do ator Gerente e ainda possui o Caso de Uso do gerenciamento de Rendimentos e Descontos, que só ele tem acesso.

As funcionalidades propostas pelos Casos de Uso foram desenvolvidas e são apresentadas no tópico 3.2.

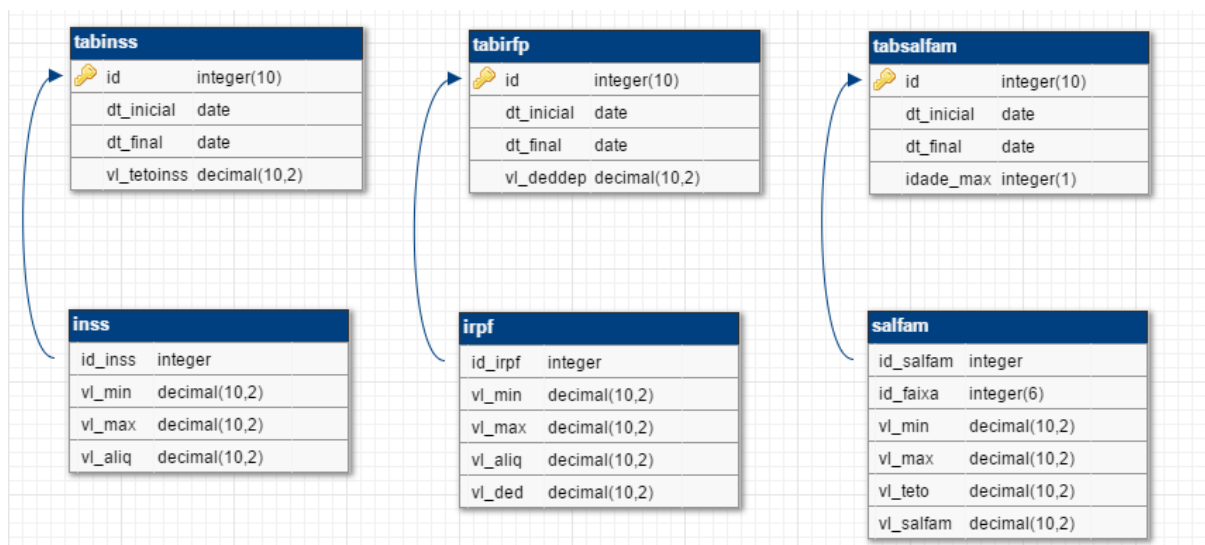
- Diagrama de Entidade e Relacionamento

A representação gráfica do Modelo de Entidade e Relacionamento (MER) é definida no Diagrama de Entidade e Relacionamento (DER). O MER é um modelo abstrato que tem como objetivo descrever, de maneira conceitual, os dados a serem utilizados em *software*. Este modelo é uma representação da realidade, composto por entidades, relacionamentos e atributos.

Neste diagrama, as entidades são representadas por retângulos: na parte superior, encontra-se o nome da entidade; já na parte interna, são descritos os seus atributos. As linhas que ligam as entidades representam os relacionamentos entre elas. Cada entidade é uma tabela e seus atributos são as colunas.

A criação do DER desta pesquisa foi elaborada em módulos, para o melhor entendimento do desenvolvedor da aplicação. A seguir, cada módulo será apresentado e explicado, juntamente com seu diagrama. A Figura 2 apresenta o primeiro módulo do DER do sistema desta pesquisa.

Figura 2 – Diagrama de Entidade e Relacionamento – Módulo I.



Fonte: Autoria própria.

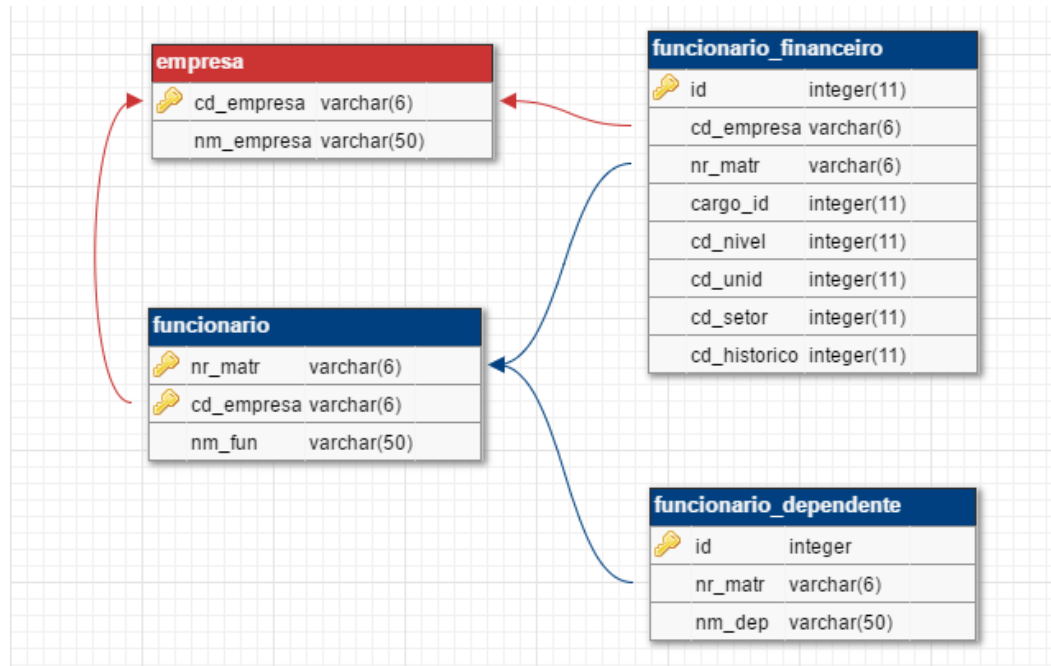
O primeiro módulo do DER mostra as tabelas relacionadas ao gerenciamento dos valores das Tabelas de INSS, IRFP e Salário-família. Como já foi dito no referencial teórico, estas tabelas possuem um prazo de vigência e são atualizadas de acordo com as ordens do governo. Para que o sistema saiba qual tabela está em vigor, foram criadas as tabelas “*tabinss*”, “*tabirpf*” e “*tabsalfam*”, nelas estão: seus respectivos identificadores, data de início e data final de vigência.

A tabela “*tabinss*” também possui o valor do teto do INSS que representa a quantia máxima a ser descontada no cálculo da previdência social. Por exemplo, quando o valor do teto do INSS é definido para R\$ 550.00, não poderá haver descontos de previdência social maior que este porte. Em “*tabirpf*”, tem-se o campo “*valordeddep*” que indica o valor a ser descontado por dependente. Em “*tabsalfam*”, tem-se a coluna “*idade_max*” que representa a idade máxima do dependente para que o funcionário tenha direito ao salário-família.

Nas tabelas “*inss*” e “*irpf*”, além do identificador da tabela vigente, encontram-se as alíquotas requeridas a partir dos limites de valores estabelecidos nas tabelas do governo. Em “*irpf*”, também está o campo “*vl_ded*” que refere-se ao valor a ser deduzido no cálculo do imposto de renda. Em “*salfam*”, encontram-se os valores do provento de salário-família a partir dos limites de salário também estabelecidos pela tabela do governo.

A Figura 3 demonstra o segundo módulo do DER do sistema desta pesquisa.

Figura 3 - Diagrama de Entidade e Relacionamento – Módulo II.



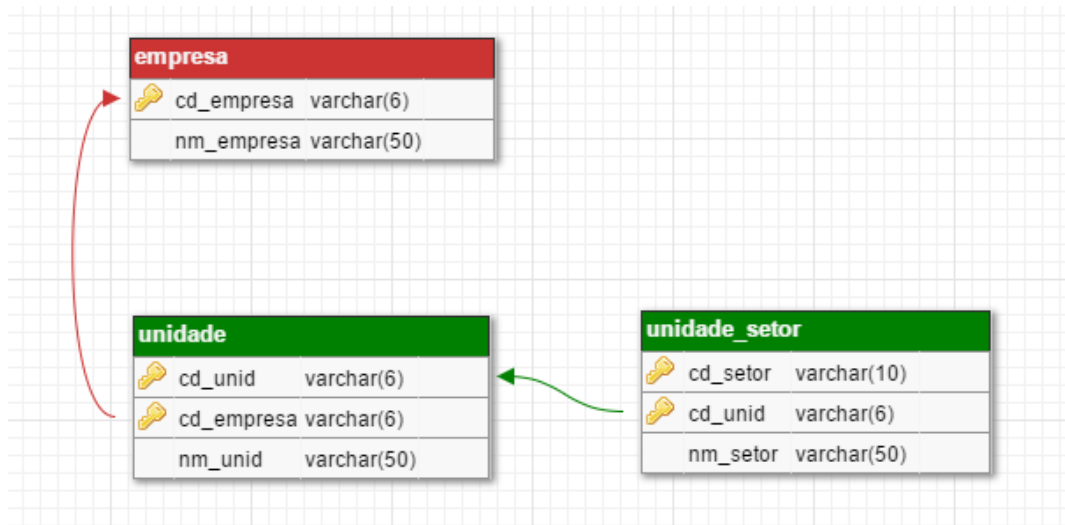
Fonte: Autoria própria.

O segundo módulo do DER é constituído pelas tabelas “*empresa*”, “*funcionario*”, “*funcionario_dependente*” e “*funcionario_financeiro*”. Em “*empresa*” estão contidas as entidades empresariais cadastradas pelos usuários do sistema, seus principais atributos são: identificador e nome. Já em “*funcionario*” são salvas as entidades que trabalham na empresa na qual o usuário encontra-se conectado ao sistema, seus principais atributos são: matrícula, código da empresa na qual ele está empregado e nome.

O Módulo II também contém a tabela “*funcionario_dependente*”, nela são armazenados os dados dos dependentes de um determinado funcionário. Os atributos mais importantes desta tabela são: identificador, nome do dependente e código do funcionário que o dependente pertence. Ainda neste módulo, tem-se a tabela “*funcionario_financeiro*”, nela são salvas as informações históricas dos funcionários de forma que o sistema possa interpretar qual o cargo e o nível que eles possuem em um determinado período.

A Figura 4 expõe o terceiro módulo do DER do sistema desta pesquisa.

Figura 4 - Diagrama de Entidade e Relacionamento – Módulo III.

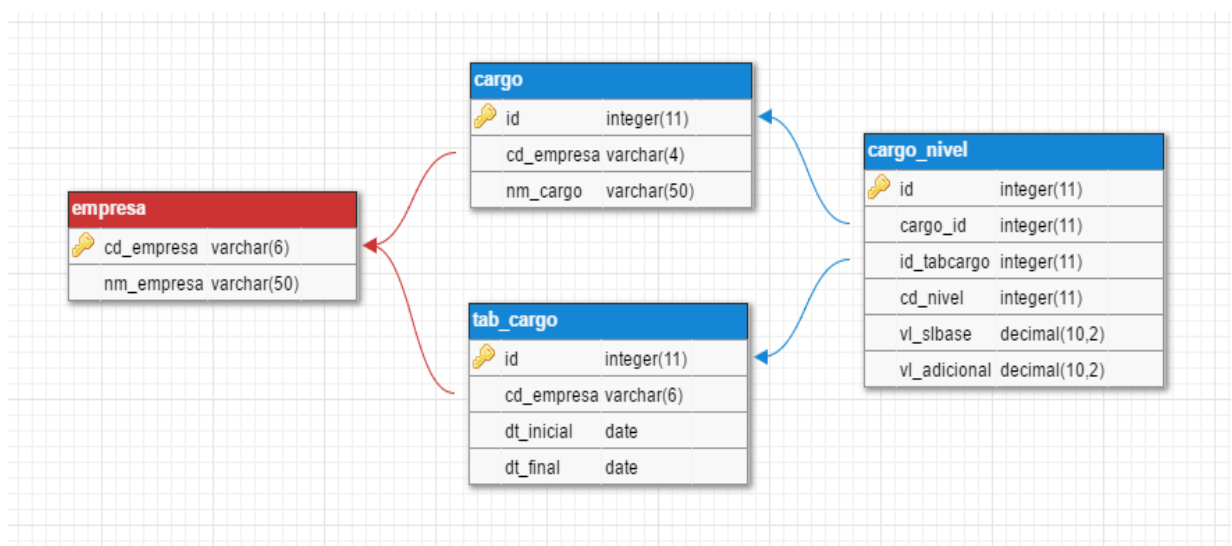


Fonte: Autoria própria.

O terceiro módulo do DER é formado pelas tabelas “*empresa*”, “*unidade*” e “*unidade_setor*”. Na tabela “*unidade*” são armazenadas as unidades de uma determinada empresa. Seus principais campos são: identificador, código da empresa que ela pertencer e nome da unidade. Já em “*unidade_setor*” estão os setores das unidades cadastradas. Esta tabela é constituída pelos campos: identificador, código da unidade a qual ela pertence e nome do setor.

A Figura 5 mostra o quarto módulo do DER do sistema desta pesquisa.

Figura 5 - Diagrama de Entidade e Relacionamento – Módulo IV.



Fonte: Autoria própria.

O quarto módulo do DER é composto pelas tabelas “*empresa*”, “*cargo*”, “*cargo_nivel*” e “*tab_cargo*”. Em “*cargo*” estão armazenados os cargos de uma determinada

empresa, seus principais campos são: identificador, o código da empresa na qual ele pertence e o nome do cargo.

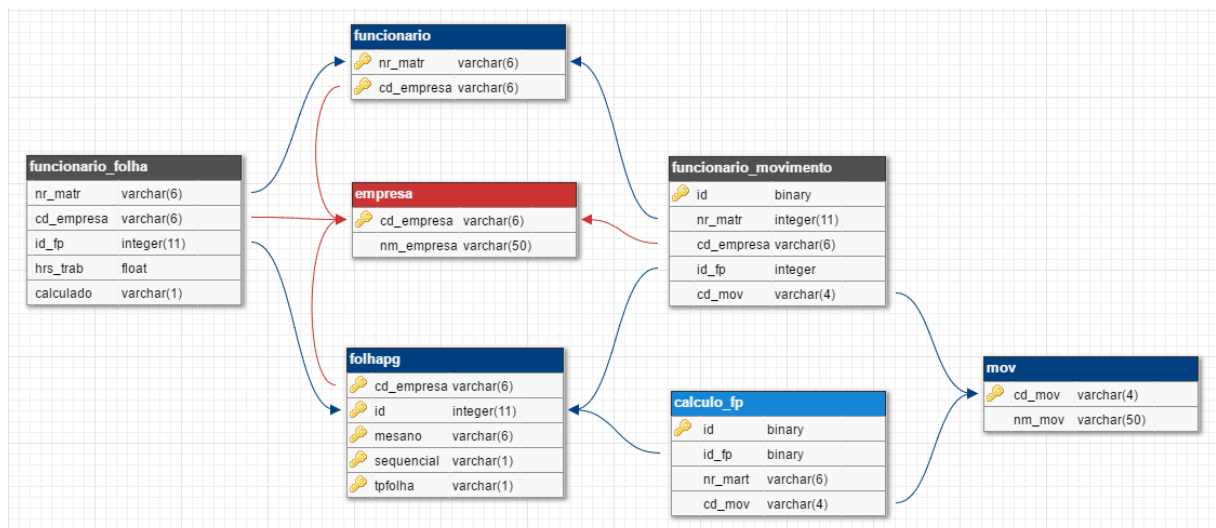
Cada cargo possui um intervalo de tempo para indicar o seu valor salarial vigente, ou seja, é possível armazenar o valor pago a cada cargo em uma determinada época. Para que esta situação seja possível, a tabela “*tab_cargo*” foi criada, nela estão contidos os campos: identificador, código da empresa que ele pertence e os intervalos de datas que indicam a validade do salário de um determinado cargo.

O Módulo IV também possui a tabela “*cargo_nivel*”, nela são armazenados os níveis dos cargos cadastrados na aplicação; seus atributos mais relevantes são: identificador, o código do cargo, o código de validade do cargo, o número do nível, o salário a ser pago para este nível e o eventual valor adicional.

Para exemplificar, tem-se: um determinado cargo tem, em seu nível inicial, um salário de R\$ 800.00 dentro do prazo de 01/01/2016 a 31/12/2016. Sendo assim, todos os funcionários que tinham este cargo nesta época, tinham como salário R\$ 800.00.

A Figura 6 mostra o quinto módulo do DER do sistema desta pesquisa.

Figura 6 - Diagrama de Entidade e Relacionamento – Módulo V.



Fonte: Autoria própria.

O quinto módulo do DER apresenta as tabelas envolvidas no cálculo da folha de pagamento: “*empresa*”, “*funcionario*”, “*folhapg*”, “*funcionario_folha*”, “*mov*”, “*funcionario_movimento*” e “*calculo_fp*”. Ao criar uma folha, suas informações são inseridas na tabela “*folhapg*”, na qual deve conter os dados para os campos principais: identificador da folha, código da empresa, mês e ano, sequencia, tipo da folha (mensal, quinzenal, semanal), unidade, setor e cargo. Durante a criação da folha, os funcionários selecionados pelos campos

unidade, setor e cargo são inseridos na tabela “*funcionario_folha*”, que tem como campos principais: matrícula do funcionário, código da empresa, o código da folha e horas trabalhadas.

Depois que a folha é criada, o usuário pode realizar a movimentação de proventos e descontos para seus funcionários. Estas movimentações são armazenadas em “*funcionario_movimento*”, que tem como atributos: identificador, matrícula do funcionário, código da empresa, código da folha e código do movimento. Os movimentos escolhidos são listados a partir da tabela “*mov*”, que contém os principais eventos e descontos cadastrados pelo usuário administrador.

Quando a folha é calculada, os resultados das movimentações são salvos em “*calcula_fp*”, ou seja, os resultados obtidos em cada provento ou desconto são armazenados nesta tabela, que tem como atributos: identificador, código da folha de pagamento, código do funcionário e código do movimento.

Os módulos dos DER ajudaram bastante durante o processo de desenvolvimento do sistema. O modelo conceitual propôs uma ideia que serviu como base para o desenvolvimento da aplicação. A seguir, as principais telas do sistema serão apresentadas e explicadas de acordo com suas funcionalidades.

3.2 SisFolha

O *SisFolha*, nome dado ao sistema desta pesquisa, tem como principal objetivo realizar o cálculo da folha de pagamento dos funcionários de uma determinada empresa. Nesta seção, as principais telas do sistema serão exibidas e explicadas conforme suas funcionalidades.

Ao entrar no sistema, é exibida para o usuário a tela de *login*, na qual ele deverá informar seu nome de usuário e senha. Caso os dados fornecidos pelo usuário sejam válidos, uma nova janela será mostrada, para que ele possa informar a empresa, o mês e o ano desejados. Logo depois que o usuário seleciona a empresa desejada, ele é direcionado para a tela inicial do sistema.

A tela inicial da aplicação contém um *menu* na parte superior com as seguintes opções: “*arquivo*”, “*cadastros*” e “*movimentos*”. Em “*cadastros*”, estão os *submenus* que encaminham o usuário para os cadastros de todas as entidades do sistema. No *menu* “*movimentos*”, encontram-se os *submenus* para Folha de Pagamento e Definição ou Mudança de Cargo/Nível.

No menu “cadastros”, encontra-se o submenu “Empresa”, que direciona o usuário para a listagem das empresas cadastradas. A Figura 7 expõe a tela de listagem e a tela de cadastro de empresa.

Figura 7 – Tela de Listagem e Tela de Cadastro de Empresa.

The image displays two side-by-side screenshots of a software application interface.

The left screenshot, titled "Cadastro de Empresas", shows a window with a search bar at the top labeled "Localizar por" and a button "F2 | Localizar". Below the search bar is a table with the following data:

Código	Razão Social	CNPJ/CEI	Telefone
001	EMPRESA TESTE 1	00.000.000/0000-00	(84) 3316-3070
003	EMPRESA TESTE 2	00.000.000/0000-00	(12) 3123-1232
004	EMPRESA TESTE 3	00.000.000/0000-00	(84) 3333-3333

At the bottom of the window is a toolbar with icons for "Insert | Incluir", "Enter | Editar", "Ctrl + Del | Excluir", and "Imprimir".

The right screenshot, titled "Cadastro Empresa", shows a window with multiple tabs: "Principal", "Classificações", "PAT", and "Contribuições". The "Principal" tab is active, showing various form fields for company registration, including "Codigo", "Razão Social", "Tipo de Inscrição" (with radio buttons for CNPJ, CEI, and CPF), "Nome Fantasia", "Inscrição Estadual (IE)", "Inscrição Municipal (IM)", "Responsável", "Contador", and "Endereço / Contato" (with fields for Logradouro, Número, Complemento, CEP, Bairro, UF, and Município). There are also fields for "Telefone", "Celular", "Fax", and "Email". At the bottom right are "OK" and "Cancel" buttons.

Fonte: Autoria própria.

Na tela da esquerda, tem-se a listagem das empresas cadastradas na aplicação. Na lista, têm-se os seguintes atributos das empresas: código, nome, CNPJ/CEI e telefone de contato. A tela possui as opções para inclusão de uma nova empresa, edição de uma entidade cadastrada, exclusão de e impressão dos dados de uma determinada organização. Ainda é possível realizar a busca de empresa pelo seu identificador ou nome.

Ao clicar na opção de inclusão de uma nova empresa, o usuário é encaminhado para a tela que possui o formulário de cadastro de empresa. Esta tela encontra-se ao lado direito da Figura 7. Nela, o usuário pode cadastrar uma nova empresa, por meio do preenchimento do formulário com os dados requisitados. Após o cadastro, a empresa passa a incorporar a listagem de empresas.

No menu “cadastros”, encontra-se também o submenu “Funcionário”, que direciona o usuário para a listagem dos funcionários já cadastrados. A Figura 8 exibe a tela de listagem e a tela de cadastro de funcionários.

Figura 8 – Tela de Listagem e Tela de Cadastro de Funcionários.

Funcionários

Exibir funcionários: ☒ Ativos ☐ Afastados ☐ Desligados ☐ Todos

Filtrar por: Unidade Setor

Localizar por: F2 | Localizar

Código	Nome	Data Admissão	CPF	N° CTPS / Serie	Unidade
000008	teste 01	01/12/2016		/ -	UNIDADE 01
000009	teste 02	01/12/2016		/ -	UNIDADE 01
000010	teste 03	01/12/2016		/ -	UNIDADE 01
000011	teste 04	01/12/2016		/ -	UNIDADE 01
000012	teste 05	01/12/2016		/ -	UNIDADE 01
000013	teste 06	01/12/2016		/ -	UNIDADE 01
000014	teste 07	01/12/2016		/ -	UNIDADE 01
000015	teste 08	01/12/2016		/ -	UNIDADE 01
000016	teste 09	01/12/2016		/ -	UNIDADE 01
000017	teste 10	01/12/2016		/ -	UNIDADE 01
000018	teste 11	01/12/2016		/ -	UNIDADE 01

Incluir Editar Excluir Dependentes Imprimir

Cadastro Funcionário

Dados Pessoais | Contrato | Documentos | Complementares | Vínculo | Aviso Prévio

Matricula Nome CPF Data de Nascimento

Nacionalidade Ano de Chegada Naturalidade (UF) Município

Sexo Raça/Cor Estado Civil Grau de Instrução

Nome Pai Nome Mãe

Endereço / Contato

Logradouro Número Complemento

CEP Bairro UF Município

Telefone Celular E-Mail

OK Cancel

Fonte: Autoria própria.

Na tela da esquerda, tem-se a listagem dos funcionários cadastrados na empresa em que o usuário encontra-se conectado na aplicação. Na lista, têm-se os seguintes atributos dos funcionários: código, nome, data de admissão, CPF, número do CTPS, unidade e setor. A tela possui as opções para inclusão de um novo funcionário, edição de um empregado cadastrado, exclusão de e impressão dos dados de um determinado trabalhador. Também é possível realizar busca de funcionário pelo seu identificador, nome ou CPF.

Ao clicar na opção de inclusão de um novo funcionário, o usuário é encaminhado para a tela que possui o formulário de cadastro de funcionário. Esta tela é exposta à direita da Figura 8. Nela, o usuário pode cadastrar um novo funcionário, por meio do preenchimento do formulário com os dados requisitados. Após o cadastro, o registro do trabalhador passa a incorporar a listagem de funcionários.

O acesso ao cadastro dos dependentes do trabalhador está disponível na tela de listagem dos funcionários, na opção “dependentes”. Ao clicar nesse botão, o usuário é encaminhado para a tela de listagem dos dependentes do funcionário selecionado. A tela possui as opções para inclusão de um novo dependente, edição de um sujeito cadastrado e exclusão de um determinado dependente.

No menu “cadastros”, encontra-se o submenu “Cargos/Níveis”, que possui os submenus “Cargos” e “Níveis”. Ao clicar sobre o submenu “Cargos”, o usuário é encaminhado para a tela de listagem dos cargos. A Figura 9 exhibe a tela de listagem e a tela de cadastro de cargos.

Figura 9 – Tela de Listagem e Tela de Cadastro de Cargos.

Código	Descrição	CBO
18	CARGO 1	317105
19	CARGO 2	212420

Fonte: Autoria própria.

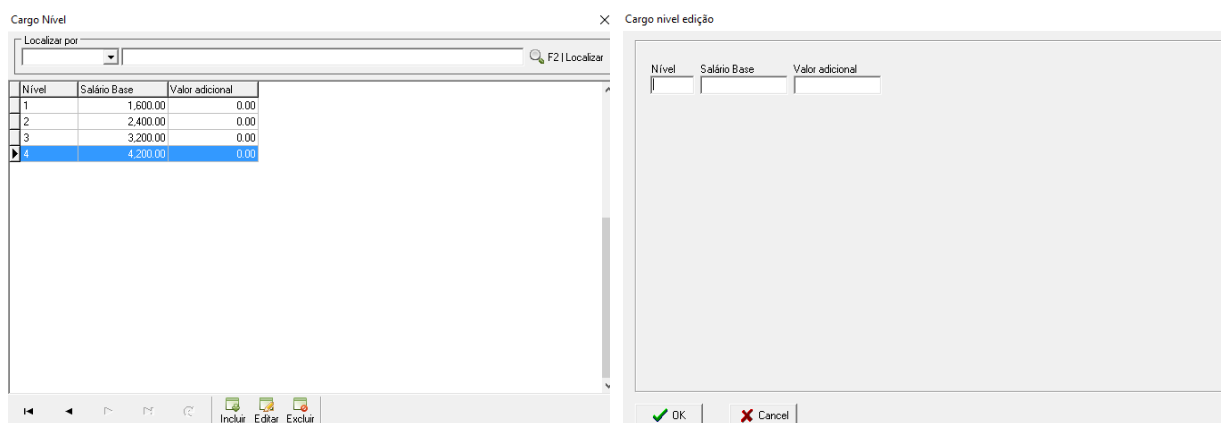
Na tela da esquerda, tem-se a listagem dos cargos cadastrados para a empresa que o usuário está conectado. Na lista, têm-se os atributos dos cargos: código, nome do cargo e o número da Classificação Brasileira de Ocupação (CBO) do cargo selecionado. O número do CBO trata-se do reconhecimento da existência de ocupações no mercado de trabalho brasileiro e é publicada pelo Ministério do Trabalho e Emprego (MTE).

A tela da esquerda também possui as opções para inclusão de um novo cargo, edição e exclusão de cargos cadastrados. Ainda é possível realizar busca de cargo pelo seu identificador ou nome. Ao clicar na opção de inclusão de um novo cargo, o usuário é deslocado para a tela que possui o formulário de cadastro de cargo. Esta tela é exibida à direita da Figura 9. Nela, o usuário pode cadastrar um novo cargo, por meio do preenchimento do formulário com os dados requisitados. Após o cadastro, o registro do cargo passa a incorporar a listagem de cargos.

Ao clicar sobre o *submenu* “Níveis”, o usuário é direcionado para a tela de listagem dos períodos de cargos e níveis, ou seja, a tela referente ao período de vigência de cargos e níveis. Nesta tela, tem-se a listagem dos períodos de vigência de cargos e níveis da empresa que o usuário está conectado. A tela também possui as opções para inclusão de um novo período, edição e exclusão de períodos cadastrados.

É possível cadastrar os níveis para os cargos já cadastrados. Na tela de listagem dos períodos vigentes tem-se o botão “Cargos”, ao clicar neste botão o usuário é encaminhado para a tela de listagem dos cargos na qual contém o botão “Níveis”. Ao clicar neste botão, o usuário é direcionado para a tela de listagem dos níveis do cargo selecionado. A Figura 10 expõe a tela de listagem e a tela de cadastro de níveis.

Figura 10 – Tela de Listagem e Tela de Cadastro de Níveis.



Fonte: Autoria própria.

Na tela da esquerda, tem-se a listagem dos níveis cadastrados para o cargo selecionado. Nesta listagem, têm-se os atributos dos níveis: número do nível, salário base e valor adicional. A tela também possui as opções para inclusão de um novo nível, edição e exclusão de níveis cadastrados. Ainda é possível realizar busca de nível pelo número de cadastro.

Ao clicar na opção de inclusão de um nível, o usuário é realocado para a tela que possui o formulário de cadastro de nível. Esta tela é apresentada à direita da Figura 10. Nela, o usuário pode cadastrar um novo nível, por meio do preenchimento do formulário com os dados requisitados. Após o cadastro, o registro do nível passa a incorporar a listagem dos níveis.

Na última opção do *menu* “cadastros”, tem-se o *submenu* “Tabelas Auxiliares”, que possui as opções para a listagem e cadastro das tabelas de INSS, IRPF, Salário-família, e ainda rendimentos e descontos. Ao clicar sobre o *submenu* “Tabela de INSS”, o usuário é encaminhado para a tela de listagem das tabelas de INSS. Nesta, tem-se a listagem das tabelas vigentes de INSS que possui os atributos: código da tabela, data inicial da vigência, data final da vigência e valor do teto salarial para a tabela selecionada. A tela também possui as opções para inclusão de uma nova tabela, edição e exclusão de tabelas cadastradas.

Depois que uma tabela é inserida, é possível realizar o cadastro de seus valores. O acesso a este cadastro está disponível no botão “Valores”, localizado na tela de listagem de tabelas do INSS. Ao clicar sobre este botão, o usuário é encaminhado para a tela de listagem dos valores da tabela INSS selecionada. A Figura 11 exibe a tela de listagem e a tela de cadastro dos valores da tabela de previdência social.

Figura 11 – Tela de Listagem e Tela de Cadastro dos valores da Tabela de INSS.

Valor Mínimo	Valor Máximo	Alíquota (%)
0,00	1.659,38	8,00
1.659,39	2.765,66	9,00
2.765,67	5.531,31	11,00

Fonte: Autoria própria.

Na tela da esquerda, tem-se a listagem dos valores da tabela de INSS selecionada. Nesta listagem, têm-se os atributos: valor mínimo, valor máximo e valor da alíquota. A tela também possui as opções para inclusão de um novo valor, edição e exclusão de valores cadastrados.

Ao clicar na opção de inclusão de um valor, o usuário é direcionado para a tela que possui o formulário de cadastro de valor. Esta tela é apresentada à direita da Figura 11. Nela, o usuário pode cadastrar um novo valor, por meio do preenchimento do formulário com os dados requisitados. Após o cadastro, o registro da tabela passa a incorporar a listagem dos valores da tabela de INSS requisitada.

Ainda no menu “cadastros”, no submenu “Tabelas Auxiliares”, tem-se a opção “Tabela de IRPF”, ao clicar nesta alternativa, o usuário é encaminhado para a tela de listagem das tabelas de IRPF. Nesta tela, tem-se a listagem das tabelas vigentes de IRPF que possui os atributos: código da tabela, data inicial da vigência, data final da vigência e valor da dedução por dependente. A tela também possui as opções para inclusão de uma nova tabela, edição e exclusão de tabelas cadastradas.

Depois que a tabela é inserida, é possível realizar o cadastro de seus valores. O acesso a este cadastro está disponível no botão “Valores”, localizado na tela de listagem de tabelas do IRPF. Ao clicar sobre este botão, o usuário é encaminhado para a tela de listagem dos valores da tabela IRPF selecionada. A Figura 12 exhibe a tela de listagem e a tela de cadastro dos valores da tabela de imposto de renda.

Figura 12 – Tela de Listagem e Tela de Cadastro dos valores da Tabela de IRPF.

The image shows two side-by-side windows from a software application. The left window, titled 'Tabela de IRPF', contains a table with the following data:

Valor Mínimo	Valor Máximo	Alíquota (%)	Dedução
0,00	1.999,18	0,00	0,00
1.999,19	2.967,98	7,50	149,94
2.967,99	3.938,60	15,00	334,80
3.938,61	4.897,91	22,50	667,94
4.897,92	9.999.999,00	27,50	912,83

The right window, titled 'Edição IRPF', is a form for editing or registering values. It has four input fields labeled 'Valor Mínimo', 'Valor Máximo', 'Alíquota (%)', and 'Dedução'. At the bottom of the right window are 'OK' and 'Cancel' buttons.

Fonte: Autoria própria.

Na tela da esquerda, tem-se a listagem dos valores da tabela de IRPF selecionada. Nesta listagem, têm-se os atributos: valor mínimo, valor máximo, valor da alíquota e valor da dedução. A tela também possui as opções para inclusão de um novo valor, edição e exclusão de valores cadastrados.

Ao clicar na opção de inclusão de um valor, o usuário é direcionado para a tela que possui o formulário de cadastro de valor. Esta tela é apresentada à direita da Figura 12. Nela, o usuário pode cadastrar um novo valor, por meio do preenchimento do formulário com os dados requisitados. Após o cadastro, o registro da tabela passa a incorporar a listagem dos valores da tabela de IRPF requisitada.

Também no *menu* “cadastros”, no *submenu* “Tabelas Auxiliares”, tem-se a opção “Tabela do Salário-família”, ao clicar nesta alternativa, o usuário é realocado para a tela de listagem das tabelas de salário-família. Nesta tela, tem-se a listagem das tabelas vigentes do salário-família que possui os atributos: código da tabela, data inicial da vigência, data final da vigência e idade máxima do dependente. A tela também possui as opções para inclusão de uma nova tabela, edição e exclusão de tabelas cadastradas.

Depois que uma tabela é inserida, é possível realizar o cadastro de seus valores. O acesso a este cadastro está disponível no botão “Valores”, localizado na tela de listagem de tabelas do salário-família. Ao clicar sobre este botão, o usuário é encaminhado para a tela de listagem dos valores da tabela de salário-família selecionada. A Figura 13 exibe a tela de listagem e a tela de cadastro dos valores da tabela de salário-família.

Figura 13 – Tela de Listagem e de Cadastro dos valores da Tabela de Salário-Família.

Faixa	Valor Mínimo	Valor Máximo	Valor Teto	Salário Família
1	0,00	806,80	1.212,64	41,37
2	806,81	1.212,64	1.212,64	29,16
3	1.212,64	9.999.999,00	1.212,64	

Fonte: Autoria própria.

Na tela da esquerda, tem-se a listagem dos valores da tabela de salário-família selecionada. Nesta listagem, têm-se os atributos: número da faixa, valor mínimo, valor máximo, valor do teto e valor do salário-família. A tela também possui as opções para inclusão de um novo valor, edição e exclusão de valores cadastrados.

Ao clicar na opção de inclusão de um valor, o usuário é direcionado para a tela que possui o formulário de cadastro de valor. Esta tela é apresentada à direita da Figura 13. Nela, o usuário pode cadastrar um novo valor, por meio do preenchimento do formulário com os dados requisitados. Após o cadastro, o registro da tabela passa a incorporar a listagem dos valores da tabela de salário-família requisitada.

Essas foram as principais tela de cadastro do *SisFolha*. Na próxima etapa, serão explicados os métodos para o cálculo da folha de pagamento.

3.3 Demonstrativo da metodologia de cálculo da folha

O cálculo da folha reúne todos os proventos e descontos de cada funcionário e gera um valor líquido, no qual é a quantia paga ao trabalhador. O método adotado para a realização do cálculo da folha deve obedecer à seguinte ordem de raciocínio:

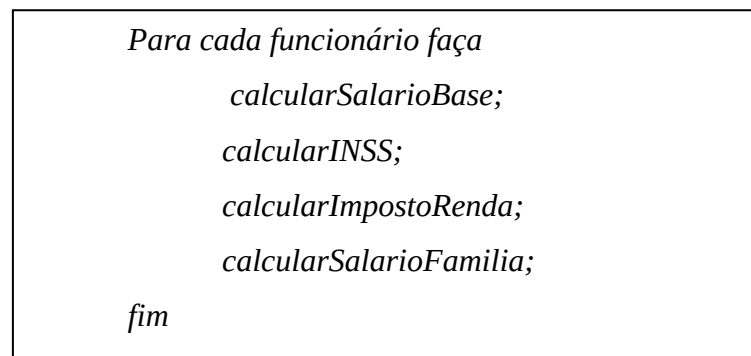
1. Calcular o Salário Base;
2. Calcular os Descontos de INSS;
3. Calcular os Descontos de IRPF;
4. Calcular o Salário-família.

A seguir, serão apresentados os métodos para o cálculo de folha.

3.3.1 Método Iterativo

O Método Iterativo realiza o cálculo da folha de pagamento percorrendo todos os funcionários, realizando os cálculos de proventos e descontos para cada um deles, ou seja, para cada funcionário calcula-se: salário base, desconto de INSS, desconto de IRPF e acréscimo de salário-família. A Figura 31 apresenta o algoritmo básico desse método.

Figura 14 – Método Iterativo.



Fonte: Autoria própria.

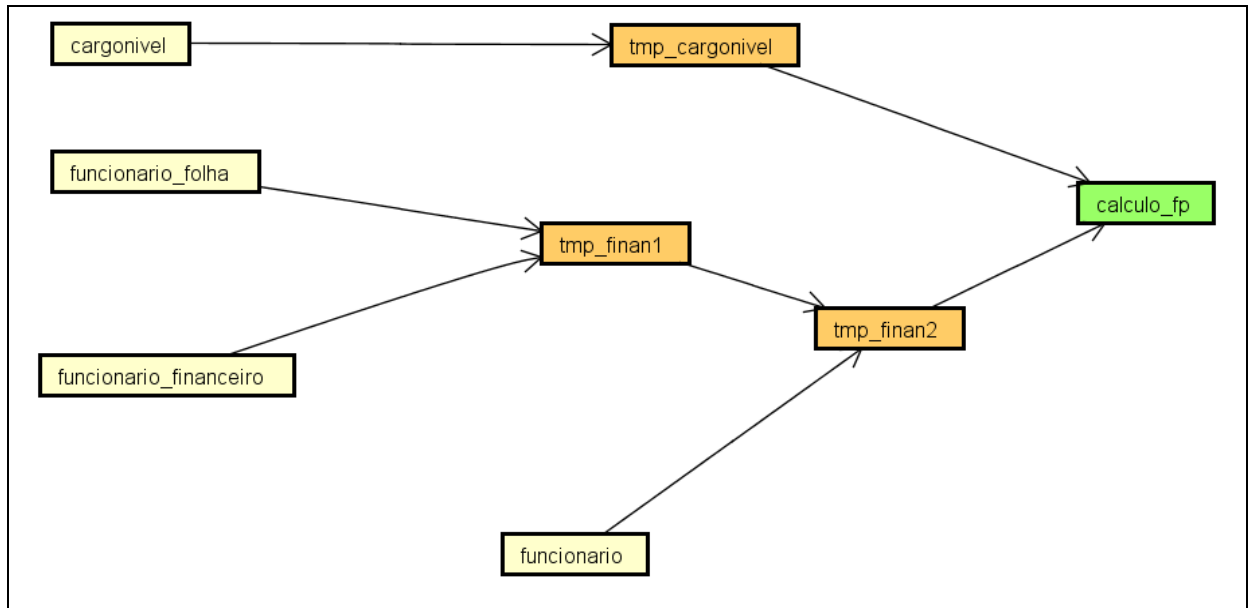
Esta abordagem gera um grande fluxo de pequenas instruções SQL sendo executadas, repetidamente, pelo servidor, causando gargalo, onerando o custo computacional e aumentando o tempo de finalização do cálculo. Apesar desse método resolver o problema, ele não será utilizado devido aos motivos já apresentados. Pretende-se ter como alternativa uma abordagem mais eficiente que realize o mesmo cálculo usando um único bloco de instruções SQL. Esta nova abordagem será explicada a seguir.

3.3.2 Método SQL de bloco único

A ideia é realizar o cálculo dos proventos e descontos de uma só vez para todos os funcionários, ou seja, uma enorme *query* é montada, como se fosse um bloco único, mantendo a ideia da refatoração das consultas do Método Iterativo, obtendo o mesmo resultado. A seguir serão apresentados os Diagramas de Fluxo de Dados para cada provento e desconto, assim como a respectiva implementação.

A Figura 16 mostra o Diagrama de Fluxo para calcular o salário base.

Figura 16 – Diagrama de Fluxo de Dados para o cálculo do Salário Base.



Fonte: Autoria própria.

O diagrama acima apresenta as tabelas envolvidas para calcular o salário base dos funcionários. Primeiramente, cria-se a tabela temporária “*tmp_cargonivel*” que contém cargos para o período atual, com os respectivos valores de cada cargo e nível. Em seguida, cria-se a tabela temporária “*tmp_finan1*” que possui os funcionários da folha em questão. Depois, cria-se a tabela temporária “*tmp_finan2*” que tem informações admissionais de cada funcionário, tais como: cargo, nível, data de admissão dentre outros. Por fim, insere-se em “*calculo_fp*” o valor do salário base de cada funcionário.

A Figura 17 exibe o código referente ao Diagrama de Fluxo de dados para calcular o salário base dos trabalhadores.

Figura 17 - Comandos SQL para calcular o salário base.

```

drop table if exists tmp_cargonivel;
create temporary table if not exists tmp_cargonivel
select c.* from cargonivel c
inner join (select * from tab_cargo where cd_empresa = @cdempresa
            and @dtref between dt_inicio and dt_final) t on t.id = c.tabcargo_id
;

drop table if exists tmp_finan1;
create temporary table if not exists tmp_finan1
select
    ff.dt_lancamento, f.cd_empresa, f.nr_matr, max(ff.id) as id
from funcionario_folha f
left join funcionario_financeiro ff on ff.nr_matr = f.nr_matr
where ff.dt_lancamento <= @dtref
and f.id_fp = @idfp
group by f.cd_empresa, f.nr_matr;

drop table if exists tmp_finan2;
create temporary table if not exists tmp_finan2
select
    a.*, b.dt_admis,
    if(datediff(@dtref,b.dt_admis)<@dias_mes ,datediff(@dtref,b.dt_admis),@dias_mes ) as qt_dias
from funcionario_financeiro a
inner join funcionario b on b.cd_empresa=a.cd_empresa and b.nr_matr=a.nr_matr
where
a.cd_empresa=@cdempresa
AND a.dt_lancamento < @dtref
AND id IN (select id from tmp_finan1)
and a.cd_unid = @cdunid
and a.cd_setor = @cdsetor
and a.cargo_id = @cdcargo
HAVING dt_admis <= @dtref;

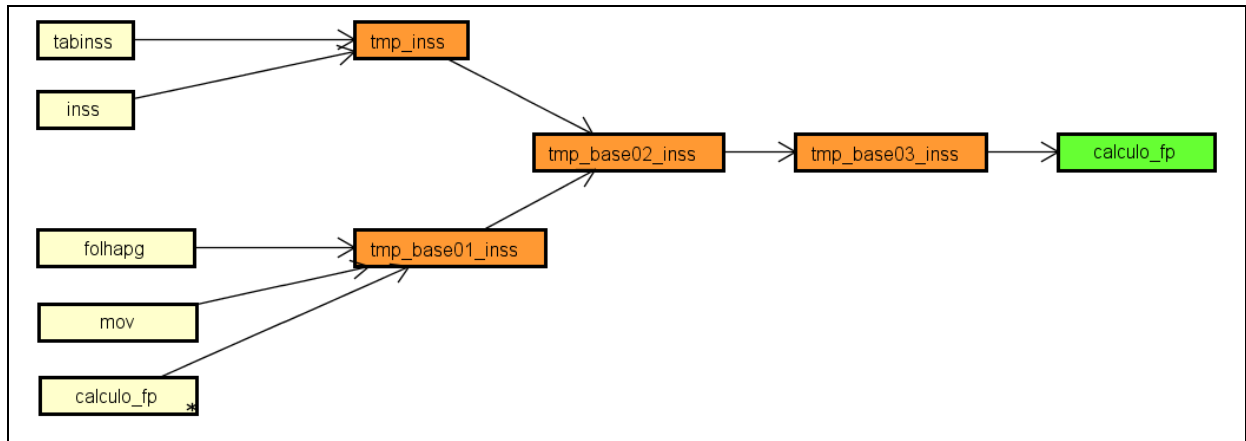
insert into calculo_fp (id_fp,nr_matr,cd_mov,vl_rendimento,vl_desconto,referencia)
select @idfp, a.nr_matr, "1001",
    ifnull(b.vl_slbase*(a.qt_dias/@dias_mes ),0) as vl_rendimento,
    0 as vl_desconto, a.qt_dias
from
    tmp_finan2 a
left join tmp_cargonivel b on b.cargo_id = a.cargo_id and b.cd_nivel = a.cd_nivel
having vl_rendimento > 0;

```

Fonte: Autoria própria.

Dando continuidade, a Figura 18 mostra o Diagrama de Fluxo para calcular o desconto do INSS.

Figura 18 - Diagrama de Fluxo de Dados para o cálculo do desconto do INSS.



Fonte: Autoria própria.

O diagrama acima apresenta as tabelas envolvidas para calcular o desconto do INSS dos funcionários. Cria-se a tabela temporária “*tmp_inss*” que contém a tabela de INSS vigente. Em seguida, cria-se a tabela temporária “*tmp_base01_inss*” que contém o valor acumulado dos rendimentos usado como base de cálculo para o INSS. Depois, cria-se a tabela temporária “*tmp_base02_inss*” que armazena a alíquota aplicada sobre a base de cálculo. Na sequência, cria-se a tabela temporária “*tmp_base03_inss*”, com o valor do desconto. Por fim, insere-se em “*calcula_fp*” o valor correspondente ao desconto de INSS.

A Figura 19 exibe o código referente ao Diagrama de Fluxo de dados para calcular o desconto de INSS dos trabalhadores.

Figura 19 - Comandos SQL para calcular o desconto de INSS.

```

drop table if exists tmp_inss;
create temporary table tmp_inss
select * from inss
where id_inss = (select id from tabinss where @dtref between dt_inicial and dt_final);

select @aliqmaxInss := max(vl_aliq) from tmp_inss;

drop table if exists tmp_base01_inss;
create temporary table if not exists tmp_base01_inss
select
    id_fp, nr_matr, vl_desconto,
    sum(vl_rendimento - vl_desconto) as vl_base
from
(select c.*
from calculo_fp c
left join folhapg f on f.id = @idpf
inner join mov e on e.cd_mov = c.cd_mov and e.sn_inss = 1
where id_fp = @idfp
and nr_matr = @func
) b group by nr_matr;

drop table if exists tmp_base02_inss;
create temporary table if not exists tmp_base02_inss
select
    a.*, b.vl_min, b.vl_max,
    if(a.vl_base between vl_min and vl_max, b.vl_aliq, @aliqmaxInss) as vl_aliq,
    round(IF(vl_base > @vl_tetoinss, @vl_tetoinss, vl_base), 2) as vl_baseinss
from tmp_base01_inss a
left join tmp_inss b on a.vl_base between vl_min and vl_max;

drop table if exists tmp_base03_inss;
create temporary table if not exists tmp_base03_inss
select a.*, round(vl_baseinss * (vl_aliq/100), 2) as vl_inss
from tmp_base02_inss a;

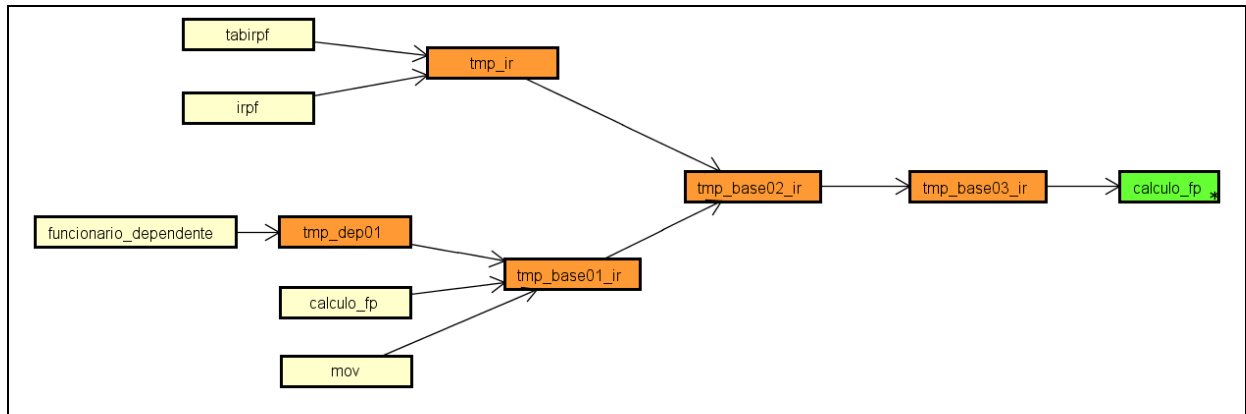
insert into calculo_fp (id_fp,nr_matr,cd_mov,vl_rendimento,vl_desconto,referencia)
select @idfp, a.nr_matr, "9100", 0, vl_inss, vl_aliq
from tmp_base03_inss a
having vl_inss > 0
[matricula];

```

Fonte: Autoria própria.

Na sequência, a Figura 20 mostra o Diagrama de Fluxo para calcular o desconto do IRPF.

Figura 20 - Diagrama de Fluxo de Dados para o cálculo do desconto de IRPF.



Fonte: Autoria própria.

O diagrama acima exhibe as tabelas envolvidas para calcular o desconto do IRPF dos funcionários. Cria-se a tabela temporária “*tmp_ir*” que contém a tabela de IRPF vigente. Depois, cria-se a tabela “*tmp_dep01*”, que possui a quantidade de dependentes dos funcionários. Em seguida, cria-se a tabela temporária “*tmp_base01_ir*” que contém o valor acumulado dos rendimentos usado como base de cálculo para o IRPF. Depois, cria-se a tabela temporária “*tmp_base02_ir*” que armazena a alíquota e o valor da dedução aplicados sobre a base de cálculo. Na sequência, cria-se a tabela temporária “*tmp_base03_ir*”, com o valor do desconto. Por fim, insere-se em “*calculo_fp*” o valor correspondente ao desconto de IRPF.

A Figura 21 exhibe o código referente ao Diagrama de Fluxo de dados para calcular o desconto de IRPF dos empregados.

Figura 21 - Comandos SQL para calcular o desconto de IRPF.

```

select @idirrf := id, @vldeducacao := vl_deddep, @qtmaximo := qt_maxdep from tabirpf
where @dtref between dt_inicial and dt_final;

drop table if exists tmp_ir;
create temporary table tmp_ir
select * from irpf where id_irpf = @idirrf;

select @aliqmaxIR := max(vl_aliq) from tmp_ir;

select @vldepmax := round(@vldeducacao * @qtmaximo, 2) as vl_depmax
from tabirpf ti where @dtref between ti.dt_inicial and ti.dt_final;

drop table if exists tmp_dep01;
create temporary table tmp_dep01
select
    nr_matr, count(id) as qt_dep
from funcionario_dependente
where 1
and sn_ir = 1
group by nr_matr;

drop table if exists tmp_base01_ir;
create temporary table if not exists tmp_base01_ir
select
    id_fp, b.nr_matr,
    round(sum(b.vl_rendimento) - sum(b.vl_desconto)
        - if((@vldeducacao * b.qt_dep) < @vldepmax, @vldeducacao * b.qt_dep, @vldepmax), 2) as vl_base
from
    (select c.*, if(d.qt_dep <> 0, d.qt_dep, 0) as qt_dep
    from calculo_fp c
    inner join mov e on e.cd_mov = c.cd_mov and e.sn_irrf = 1
    left join tmp_dep01 d on d.nr_matr = c.nr_matr
    ) b
where id_fp = @idfp
group by nr_matr;

drop table if exists tmp_base02_ir;
create temporary table if not exists tmp_base02_ir
select a.*,
    if(a.vl_base between vl_min and vl_max, b.vl_aliq, @aliqmaxIR) as vl_aliq,
    if(b.vl_ded = 0, max(b.vl_ded), b.vl_ded) as vl_ded
from tmp_base01_ir a
left join tmp_ir b on a.vl_base between vl_min and vl_max
group by a.nr_matr;

drop table if exists tmp_base03_ir;
create temporary table if not exists tmp_base03_ir
select a.*, ((vl_base) * (vl_aliq/100)) - vl_ded as vl_ir
from tmp_base02_ir a;

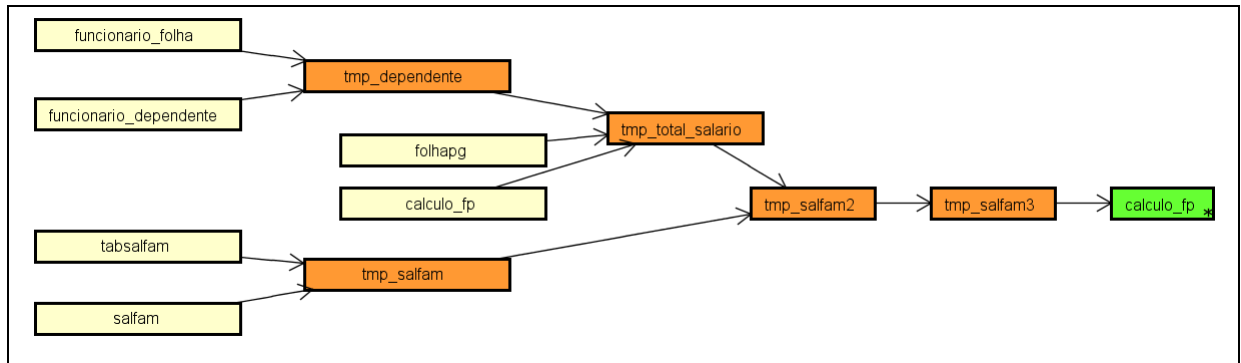
insert into calculo_fp (id_fp,nr_matr,cd_mov,vl_rendimento,vl_desconto,referencia)
select
    @idfp, a.nr_matr, "9200", 0 as vl_rendimento, vl_ir as vl_desconto, vl_aliq
from tmp_base03_ir a
where 1 having vl_ir > 0;

```

Fonte: Autoria própria.

Logo mais, a Figura 22 mostra o Diagrama de Fluxo para calcular o salário-família.

Figura 22 - Diagrama de Fluxo de Dados para o cálculo do Salário-Família.



Fonte: Autoria própria.

O diagrama acima exhibe as tabelas envolvidas para calcular o salário-família dos trabalhadores. Cria-se a tabela temporária “tmp_dependente” que possui a quantidade de dependentes dos funcionários. Depois, cria-se a tabela temporária “tmp_salfam” que contém a tabela de salário-família vigente. Em seguida, cria-se a tabela temporária “tmp_total_salario” que contém o valor acumulado dos rendimentos usado como base de cálculo para o salário-família. Depois, cria-se a tabela temporária “tmp_salfam2” que armazena o valor do salário-família definido pelo salário base dos funcionários. Na sequência, cria-se a tabela temporária “tmp_salfam3”, com o valor do salário-família. Por fim, insere-se em “calculo_fp” o valor correspondente ao provento em questão.

A Figura 23 exhibe o código referente ao Diagrama de Fluxo de dados para calcular o salário-família dos funcionários.

Figura 23 - Comandos SQL para calcular o Salário-Família.

```

SELECT @idade_maxima := idade_maxima from tabsalfam where @dtref between dt_inicial and dt_final;

drop table if exists tmp_dependente;
create temporary table if not exists tmp_dependente
select b.id_fp, b.cd_empresa, b.nr_matr, sum(b.qt_ir) as qt_ir, sum(b.qt_familia) as qt_familia from
(select @idfp as id_fp, cd_empresa, nr_matr, count(*) as qt_ir, 0 as qt_familia
from funcionario_dependente where cd_empresa = @cdempresa and sn_ir = 1
group by cd_empresa, nr_matr
union all
select @idfp as id_fp, cd_empresa, nr_matr, 0 as qt_ir, count(*) as qt_familia
from funcionario_dependente where cd_empresa = @cdempresa and sn_familia = 1
AND (TIMESTAMPDIFF(YEAR, dt_nasc, @dtref) <= @idade_maxima AND tp_depend = 2)
group by cd_empresa, nr_matr) b
inner join funcionario_folha ff on ff.nr_matr = b.nr_matr and ff.id_fp = @idfp
group by b.cd_empresa, b.nr_matr;

SELECT @id_tabsalfam := id from tabsalfam where @dtref between dt_inicial and dt_final;

drop table if exists tmp_salfam;
create temporary table if not exists tmp_salfam
select * from salfam where id_tabsalfam=@id_tabsalfam;

drop table if exists tmp_total_salario;
create temporary table if not exists tmp_total_salario
SELECT a.id_fp, f.cd_empresa, a.nr_matr, SUM(a.vl_rendimento) AS vl_salario, b.qt_familia
FROM folhag f
LEFT JOIN calculo_fp a ON f.id = a.id_fp
INNER JOIN tmp_dependente b ON b.id_fp = f.id AND b.nr_matr = a.nr_matr AND b.qt_familia > 0
WHERE f.cd_empresa = @cdempresa AND f.mesano = @mesano AND cd_mov IN ("1001","1002")
GROUP BY f.cd_empresa,a.nr_matr;

drop table if exists tmp_salfam2;
create temporary table tmp_salfam2
select a.*, b.*
from tmp_total_salario a
left join (select * from tmp_salfam) b on a.vl_salario between b.vl_min and b.vl_max;

drop table if exists tmp_salfam3;
create temporary table tmp_salfam3
select a.*, qt_familia * vl_slfam as vl_salario_familia
from tmp_salfam2 a;

insert into calculo_fp (id_fp,nr_matr,cd_mov,vl_rendimento,vl_desconto,referencia)
select @idfp, a.nr_matr, "3001", vl_salario_familia, 0 as vl_desconto, a.qt_familia
from tmp_salfam3 a
where vl_salario_familia > 0
[matricula];

```

Fonte: Autoria própria.

Por fim, tem-se o resultado de todos os proventos e descontos de cada empregado inseridos na tabela “*calculo_fp*”. A partir de então, pode-se calcular o valor líquido para cada empregado.

4. COMPARAÇÕES

Neste capítulo, será apresentado um estudo de caso elaborado a partir do sistema desta pesquisa. O estudo consiste em uma análise de desempenho do método de cálculo de folha de pagamento, para diferentes quantidades de funcionários. O estudo foi realizado, primeiramente, a partir de um banco de dados local, e posteriormente, em um banco remoto. A seguir, descreve-se o ambiente de teste utilizado.

4.1 Ambiente de Teste

Nesta seção, são apresentados o Banco de Dados e as configurações da máquina utilizadas para a realização dos testes de desempenho.

4.1.1 Configurações das Máquinas

As configurações da máquina utilizada como servidor de Banco de Dados são:

- *Notebook DELL;*
- Processador Intel® Core (TM) i5-5200U CPU 2.20GHz;
- 4.00GB de memória RAM;
- *Windows 10 Home Single Language.*

As configurações da máquina utilizada como cliente são:

- *Notebook SONY;*
- Processador Intel® Core (TM) i5-2410M CPU 2.30GHz;
- 8.00GB de memória RAM;
- *Windows 7 Home Premium.*

4.2 Resultados

Neste tópico, são apresentados os resultados obtidos a partir da execução de testes de desempenho. Estes resultados foram gerados com base no tempo de processamento gasto ao calcular uma folha de pagamento com o Método SQL. Primeiramente, são expostos os resultados para o ambiente local, em seguida, para o ambiente remoto.

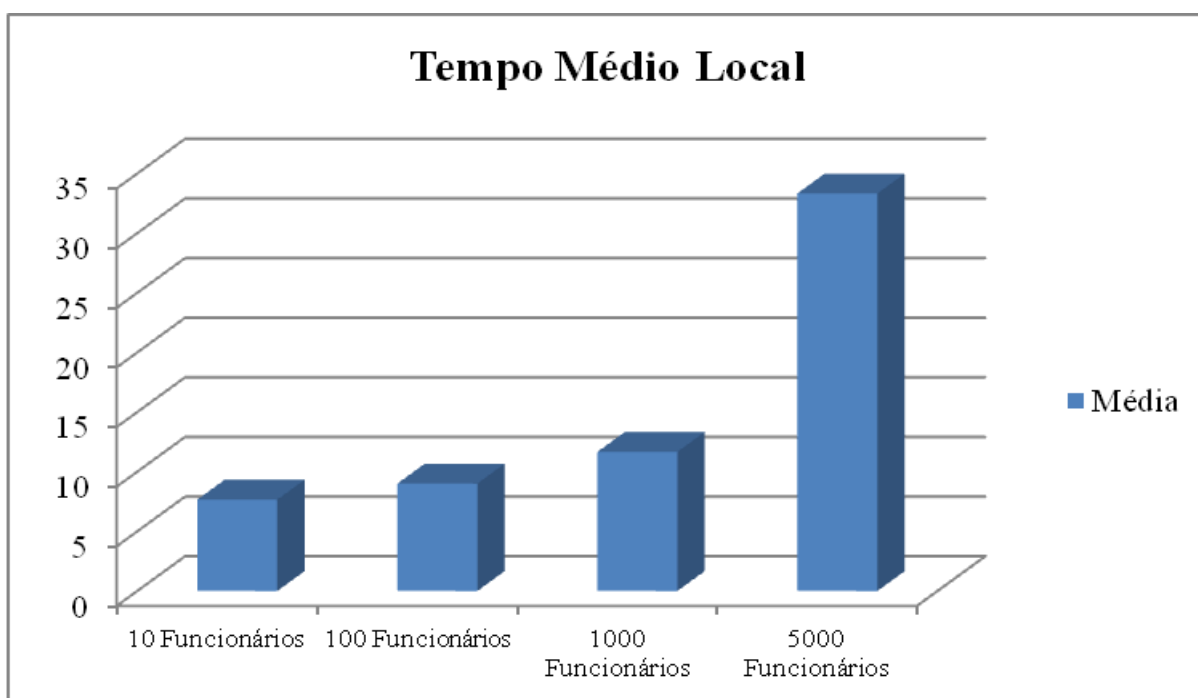
As tabelas e gráficos apresentados a seguir exibem o tempo médio do cálculo da folha de 10, 100, 1000 e 5000 funcionários, efetuados por três vezes no ambiente local e remoto. Estes elementos gráficos consideram os seguintes aspectos:

- 1 segundo = 0,000277778 horas;
- 1 minuto = 0,0166667 horas;
- 1 h = 1,0 horas.

Ainda deve-se considerar que, para a folha de 10 funcionários, todos possuem 2 dependentes, 5 recebem salário-família e 5 possuem desconto de IRPF. Na folha de 100 empregados, todos possuem 2 dependentes, 50 recebem salário-família e 50 têm desconto de IRPF. Na folha de 1000 funcionários, todos também possuem 2 dependentes, 500 recebem salário-família e 500 têm desconto de IRPF. E na folha de 5000 trabalhadores, todos possuem 2 dependentes, 2500 recebem salário-família e 2500 possuem desconto de IRPF.

O Gráfico 1 mostra os resultados relativos ao cálculo da folha no ambiente.

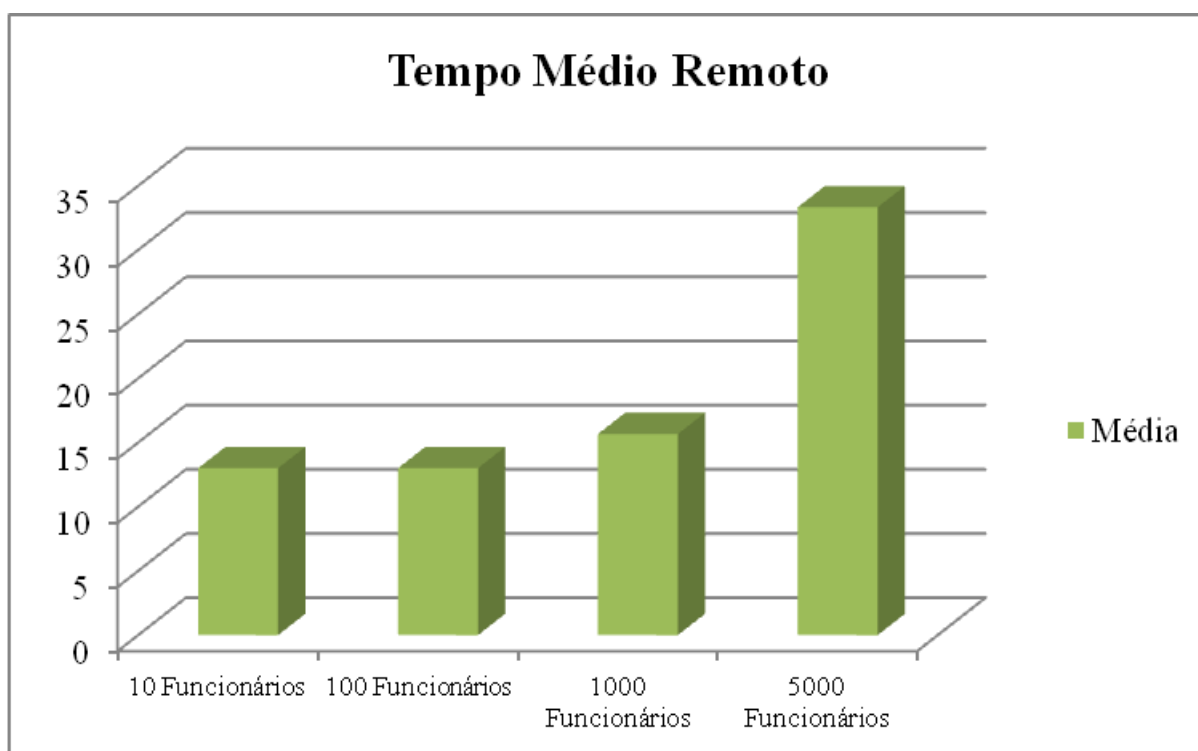
Gráfico 1 – Tempo Médio da folha de pagamento no ambiente local.



Fonte: Autoria própria.

O Gráfico 2 mostra os resultados relativos ao cálculo da folha no ambiente remoto.

Gráfico 2 – Tempo Médio da folha de pagamento no ambiente remoto.



Fonte: Autoria própria.

O Gráfico 1 mostra que o resultado do cálculo da folha para 10 funcionários é realizado no tempo médio de 7 segundos. Já para 100 funcionários, a execução do cálculo da folha é feito em aproximadamente 9 segundos. Para 1000 empregados, tem-se o tempo médio de 11 segundos. E para 5000 trabalhadores, a média da execução é de 33 segundos.

Já o Gráfico 2 mostra que o resultado do cálculo da folha para 10 funcionários é realizado no tempo médio de 13 segundos. Para 100 funcionários, a execução do cálculo da folha é feito também em 13 segundos. Para 1000 empregados, tem-se o tempo médio de 15 segundos. E para 5000 trabalhadores, a média da execução é de 33 segundos.

5. Conclusões

A modelagem de projeto de *software* foi importante, pois além de garantir que o sistema seja funcional, cumpra os seus requisitos e tenha qualidade, ela ainda se mostra indispensável para futuros projetos de evolução da aplicação. Para implementar a nova metodologia de calcular a folha de pagamento foi interessante conhecer as funcionalidades do sistema, bem como o diagrama de Entidade e Relacionamento do Banco de Dados criado. Os diagramas auxiliaram no desenvolvimento da nova abordagem, identificando o principal módulo para otimização de comandos SQL. Sendo assim, o objetivo de apresentar a modelagem do projeto de *software* e as principais funcionalidades do sistema foi atingido.

A pesquisa constatou que o fluxo básico das principais rotinas de cálculo de folha de pagamento, utilizando o Método SQL de bloco único, obteve desempenho considerável em seus resultados. Nos testes realizados com 10, 100, 1000 e 5000 funcionários, o Método SQL obteve tempo médio de 7s, para folha com 10 funcionários, no ambiente local, e 13s, no ambiente remoto; com 100 funcionários, a média foi aproximadamente de 9s, no ambiente local, e próxima de 13s, no remoto; com 1000 funcionários, a média foi aproximadamente de 11s, no ambiente local, e 15s, no remoto; com 5000 funcionários, a média foi em cerca de 33s, tanto no ambiente local quando no remoto.

Então, o objetivo de desenvolver os cálculos dos eventos utilizando instruções SQL para o sistema legado foi atendido. O Método SQL de bloco único passa os códigos para o servidor de Banco de Dados, deixando a responsabilidade de efetuar todos os cálculos de uma só vez no próprio servidor. Desta maneira, o resultado esperado pelo usuário do *SisFolha* é gerado rapidamente, mesmo que a folha de pagamento em questão tenha uma grande quantidade de funcionários.

REFERÊNCIAS BIBLIOGRÁFICAS

CAIXA. INSS. Disponível em: <<http://www.caixa.gov.br/beneficios-trabalhador/inss/Paginas/default.aspx>>. Acesso em: 14 de fevereiro de 2017.

DATE, C. J. **Introdução a sistemas de banco de dados**. Tradução de Daniel Vieira. Rio de Janeiro: Elsevier, 2003.

DIAS NETO, A. **O processo de refatoração de Banco de Dados**. Disponível em: <<http://www.devmedia.com.br/o-processo-de-refatoracao-de-banco-de-dados-artigo-revista-sql-magazine-86/20400>>. Acesso em: 13 de fevereiro de 2017.

ELMASRI, R.; NAVATHE, E.. **Sistemas de Banco de Dados**. 6. ed. Traduzido por Daniel Vieira. São Paulo: Pearson Addison Wesley, 2011.

Ferrari, M. **Tabelas temporárias no Banco de Dados Oracle**. Disponível em: <<http://www.devmedia.com.br/tabelas-temporarias-no-banco-de-dados-oracle/1915>>. Acesso em: 27 de abril de 2017.

GUIA TRABALHISTA. **Tabela do Salário-família**. Disponível em: <http://www.guiatrabalhista.com.br/guia/salario_familia.htm>. Acesso em: 28 de março de 2017.

GONÇALVES, G. **Resumo prático de Folha de Pagamento**. Editora Juruá, 4ª Edição, 2007.

HEUSER, C. A. **Projeto de Banco de Dados**. Editora Bookman, 6ª Edição, 2009.

KORTH, H. F.; SILBERSCHATZ, A.; SUDARSHAN, S. **Sistema de Banco de Dados**. Tradução de Daniel Vieira. Rio de Janeiro: Elsevier, 2006.

LEANDRO, D. **Arquitetura Client/Server e Three-tier**. Disponível em: <<http://www.devmedia.com.br/arquitetura-client-server-e-three-tier/5865>>. Acesso em: 27 de abril de 2017.

LINS, J. **Análise de paradigmas de Bancos de Dados a partir do desenvolvimento de um sistema para assistência de enfermagem no rastreamento de câncer ginecológico**. Monografia (Graduação em Ciência da Computação), Departamento de Ciências Exatas e Naturais, Universidade Federal Rural do Semi-árido, Mossoró-RN, 2016.

MACHADO, F. N. R. **Banco de Dados Projeto e Implementação**. Editora Érica, 2ª Edição, 2008.

MYSQL. Site oficial. Disponível em: <
<http://mysql.localhost.net.ar/downloads/mysql/4.1.html>>. Acesso em: 20 de maio de 2017.

MOREIRA, W. **INNER, CROSS, LEFT, RIGTH E FULL JOINS**. Disponível em: <
<http://www.devmedia.com.br/inner-cross-left-rigth-e-full-joins/21016>>. Acesso em 27 de abril de 2017.

OLIVEIRA, E. **Estudo de Caso**. Disponível em:
 <<http://www.infoescola.com/sociedade/estudo-de-caso/>>. Acesso em: 8 de janeiro de 2017.

PORTAL TRIBUTÁRIO. **Imposto de Renda**. Disponível em:
 <<http://www.portaltributario.com.br/tributario/impostoderenda.htm>>. Acesso em: 14 de fevereiro de 2017.

PREVIDÊNCIA. **Salário-família**. Disponível em: <<http://www.previdencia.gov.br/servicos-ao-cidadao/todos-os-servicos/salario-familia/>>. Acesso em: 14 de fevereiro de 2017.

PREVIDÊNCIA. **Tabela de contribuição mensal**. Disponível em:
 <<http://www.previdencia.gov.br/servicos-ao-cidadao/todos-os-servicos/gps/tabela-contribuicao-mensal>>. Acesso em: 27 de março de 2017.

RECEITA. **IRFP (Imposto sobre a Rendas das Pessoas Físicas)**. Disponível em: <
<http://idg.receita.fazenda.gov.br/acesso-rapido/tributos/irpf-imposto-de-renda-pessoa-fisica>>. Acesso em: 28 de março de 2017.

SANTOS, J. **Escrevendo queries otimizadas no SQL Server**. Disponível em: <
<http://www.devmedia.com.br/escrevendo-queries-otimizadas-no-sql-server/20403>>. Acesso em: 13 de fevereiro de 2017.